(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 October 2001 (25.10.2001)

PCT

(10) International Publication Number
WO 01/80067 A1

(51) International Patent Classification[7]: G06F 17/00, 15/16

(21) International Application Number: PCT/US01/08265

(22) International Filing Date: 14 March 2001 (14.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/550,348 14 April 2000 (14.04.2000) US

(71) Applicant: **YODLEE.COM, INC.** [US/US]; Suite 200, 3600 Bridge Parkway, Redwood Shores, CA 94065 (US).

(72) Inventors: **RANGARAJAN, Anand**; 450 N. Mathilda Avenue #1-205, Sunnyvale, CA 94086 (US). **LEE, Ji, Hoon**; 1432 Ridgeback RD #B, Chula Vista, CA 91910 (US). **INALA, Suman, Kumar**; Apartment 154, 3707 Poincianna Drive, Santa Clara, CA 95051 (US). **SATYAVOLU, Ramakrishna**; Apt. 145, 3707 Poincianna Drive, Santa Clara, CA 95051 (US). **RAJAN, Sreeranga, P.**; 3475 Granada Avenue #320, Santa Clara, CA 95051 (US).

(74) Agent: **BOYS, Donald, R.**; P.O. Box 187, Aromas, CA 95004 (US).

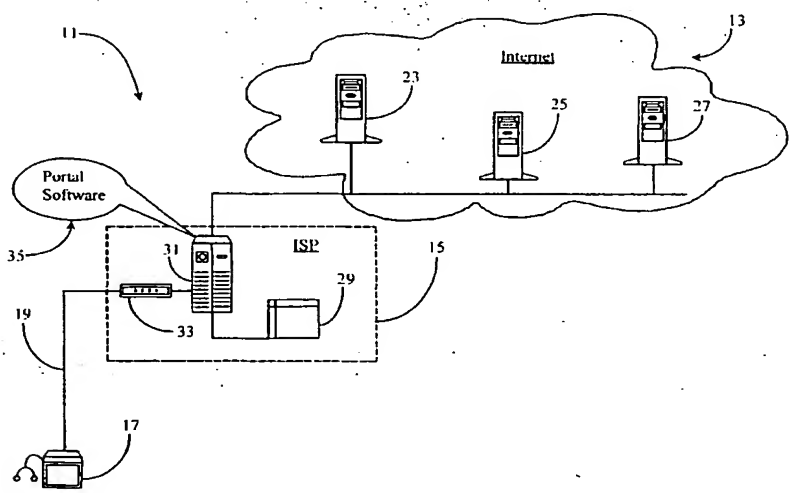
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS FOR PROVIDING AUTO-REGISTRATION AND SERVICE ACCESS TO INTERNET SITES FOR INTERNET PORTAL SUBSCRIBERS



(57) Abstract: A method and apparatus is provided for populating and submitting electronic forms by proxy over a data-packet-network (19). The apparatus comprises a software application (35) running on a system of network-connected servers (31) that enables a user (17), connected in session with one of the servers, to navigate to a site (23) containing an electronic form and obtain data about the site and about the form. The data obtained is used in conjunction with data about the user to construct a machine readable job order upon user request that may be executed for the purpose of automatic form population and submission to a host sponsoring site (15). Upon acceptance of the submitted form, data used for passwords, log-in codes and user-names is returned to a data repository where it is entered along with specific site data as a new registered site item for a registering user such that future navigation to the site, auto log-in and data may be performed automatically on behalf of the user.

WO 01/80067 A1

Method and Apparatus for Providing Auto-Registration and Service Access to Internet Sites for Internet Portal Subscribers

Field of the Invention

The present invention is in the field of Internet navigation including various communication and connection technologies and pertains more particularly to methods and apparatus, including software, for providing auto-registration services to Internet sites for portal users operating on the Internet.

Cross-Reference to Related Documents

The present invention is a continuation in part (CIP) to a patent application serial No. 09/532,647, filed 3/22/2000 entitled ***"Method and Apparatus for Retrieving Information From Semi-Structured, WEB-Based Data Sources"***, which is a continuation in part (CIP) of a patent application S/N 09/323,598 entitled ***"Method and Apparatus for Obtaining and Presenting WEB Summaries to Users"*** filed on 06/01/99, which is a continuation in part (CIP) to a patent application S/N 09/208,740 entitled ***"Method and Apparatus for Providing and Maintaining a User-Interactive Portal System Accessible via Internet or other Switched-Packet-Network"*** filed on 12/08/98, disclosures of which are incorporated herein in their entirety herein by reference.

Background of the Invention

The information network known as the World Wide Web (WWW), which is a subset of the well-known Internet, is arguably the most complete source of publicly accessible information available. Anyone with a suitable Internet appliance such as a personal computer with a standard Internet connection may access (go on-line) and navigate to information pages (termed web pages) stored on Internet-connected

servers for the purpose of garnering information and initiating transactions with hosts of such servers and pages.

Many companies offer various subscription services accessible via the Internet. For example, many people now do their banking, stock trading, shopping, and so forth from the comfort of their own homes via Internet access. Typically, a user, through subscription, has access to personalized and secure WEB pages for such functions. By typing in a user name and a password or other personal identification code, a user may obtain information, initiate transactions, buy stock, and accomplish a myriad of other tasks.

One problem that is encountered by an individual who has several or many such subscriptions to Internet-brokered services is that there are invariably many passwords and/or log-in codes to be used. Often a same password or code cannot be used for every service, as the password or code may already be taken by another user. A user may not wish to supply a code unique to the user such as perhaps a social security number because of security issues, including quality of security, that may vary from service to service. Additionally, many users at their own volition may choose different passwords for different sites so as to have increased security, which in fact also increases the number of passwords a user may have.

Another issue that can plague a user who has many passworded subscriptions is the fact that they must bookmark many WEB pages in a computer cache so that they may quickly find and access the various services. For example, in order to reserve and pay for airline travel, a user must connect to the Internet, go to his/her book-marks file and select an airline page. The user then has to enter a user name and password, and follow on-screen instructions once the page is delivered. If the user wishes to purchase tickets from the WEB site, and wishes to transfer funds from an on-line banking service, the user must also look for and select the personal bank or account page to initiate a funds transfer for the tickets. Different user names and passwords may be required to access these other pages, and things get quite complicated.

Although this preceding example is merely exemplary, it is generally known that much work related to finding WEB pages, logging in with passwords, and the like is required to successfully do business on the WEB.

A service known to the inventor and described in disclosure referenced by S/N 09/208,740 listed under the cross-reference to related documents section provides a WEB service that allows a user to store all of his password protected pages in one location such that browsing and garnering information from them is much simplified. A feature of the above service allows a user to program certain tasks into the system such that requested tasks are executed by an agent (software) based on user instruction. The service stores user password and log-in information and uses the information to log-in to the user's sites, thus enabling the user to navigate without having to manually input log-in or password codes to gain access to the links.

The above-described service uses a server to present a user-personalized application that may be displayed as an interactive home page that contains all of his listed sites (hyperlinks) for easy navigation. The application lists the user's URLs in the form of hyperlinks such that a user may click on a hyperlink and navigate to the page wherein login, if required, is automatic, and transparent to the user.

The application described above also includes a software agent that may be programmed to perform scheduled tasks for the user including returning specific summaries and updates about user-account pages. A search function is provided and adapted to cooperate with the software agent to search user-entered URLs for specific content if such pages are cached somewhere in their presentable form such as at the portal server, or on the client's machine.

A further enhancement to the system described above is known to the inventor and described in the disclosure of application S/N 09/323,598 also listed under the cross-reference section. The described enhancement consists of a means for obtaining information from WEB-based sources using a site-navigation script, a field template, and a means for parsing data. The navigation script follows site logic of a target WEB site containing the data for return to a user. Part of the template includes the description and location of the data requested by a user. A parsing engine acts to

identify the new data for retrieval for a user. In this way, WEB summaries may be compiled on updated data at user-frequented sites.

A further enhancement to the above system is known to the inventor and disclosed under the patent application listed in the cross-reference section as P3918PA and entitled ***"Method and Apparatus for Retrieving Information From Semi-Structured, WEB-Based Data Sources"***.

In the enhancement described above, an Internet search system is structured for efficient data retrieval from semi-structured data sources. Using a text selection and configuration tool, a user is enabled to navigate to a WEB-site, or multiple WEB-sites, and pre-select the block areas of information that contain data that he or she wishes to be accessed and retrieved during future automated data searches. The data type entered by a provided data input function is associated with a WEB page block selected, and upon search implementation the block selected is searched for the data type requested, and data found is retrieved to be provided to the user. In a preferred embodiment portions of the system are executed on a user station, and other portions on a Portal server to which the user may subscribe. In this way, a user may have data retrieved from semi-structured sources and presented in the form of a summary page.

The Portal system with the above-described enhancements taught in the cross-referenced patent applications also features an automatic registration service available to users if they want information from a WEB-site or service that requires registration. If the WEB pages are new to a user, and the user has no registration with the WEB site, then through agreement, or other convention, the service may be provided access to such sites. Such an agreement may be made, for example, if the host of the WEB site realizes a possibility of gaining a new customer if the customer likes the summary information presented. In disclosure taught thus far, auto registration is fairly limited to include only sites that are requested by a user and whom are cooperative to the procedure.

It has occurred to the inventor that the functions of auto-registration may be expanded to include a plurality of pre-configured WEB-services which are looking for new clients. Such a value-added service may provide researched and rated service

sites that collectively provide all of the types of singular or multiple services that a Portal subscriber may need or desire.

Auto-registration as taught thus far in applications known to the inventor still requires considerable data input from a user wishing to register to a site. Moreover, a subscriber must perform his or her own analytical research in site comparison in order to determine which of a plurality of available sites are most likely to provide optimum service to the user.

An object of the present invention is to make such services available in automated fashion allowing a user to gain such services through a novel one-touch technology that is not available with prior auto-registration services. Another object of the present invention is to rate such services based on performance and to select, and in some cases customize, such services on behalf of users according to profile information stored on behalf of users.

What is clearly needed is a method and apparatus that enables a user to request and receive automatic registration to WEB sites or services through a one-touch function from a single user interface. Such a system will provide an effective value-added service to users wherein most pre-registration complexities associated with weighted decisions regarding which service to choose from and what service options to obtain from a service may be eliminated.

Summary of the Invention

According to an embodiment of the present invention, a software application is provided for populating and submitting interactive forms by proxy. The application comprises a function for finding and capturing data about a site associated with the form and about the form associated with the site, a function for writing a job order containing data specific to the site, the associated form, and a requesting user; a function for navigating to the site and submitting data to a host sponsoring the site using the form associated with the site, and a function for returning and recording a portion of the form-submitted data accepted by the host for subsequent use in gaining access to the site.

In one aspect of the present invention the site is an interactive information-page located on a data-packet-network. In another aspect, the site is an electronic WEB page and the data-packet-network is the Internet network. In preferred embodiments, the form is an interactive registration form contained within the site. In other embodiments, the form may be a purchasing form, a temporary registration form, a survey form, or other types of interactive forms.

The application functions to enable users to obtain services by proxy from a host company operating on the network wherein the host is a company providing a service available through the site and accessible through the form or forms contained in the site.

In another aspect of the invention all of the software functions available to the application are resident in a single server-node on the Internet and are executed from the single server-node. In a preferred aspect the software functions are distributed over a plurality of server nodes connected to the Internet and are executed from those server nodes. The job order directing the navigation, form filling, and form submission is in preferred applications, written in the form of eXtensible mark-up language. In other aspects, other machine-readable languages may be incorporated.

According to still another aspect of the present invention, a hardware-software system is provided for configuring and executing a job order for populating and submitting an interactive form to a host and returning data from the host. The system includes a portal server running configuration software for interfacing with a user and for accepting data input in the form of an order request and configuration data, a generation sever running order-writing software for generating the job order according to a job request initiated by the portal server, a navigation server running form population software for executing the job order received from the generation server according to instruction and data contained therein, and a data repository accessible to the servers for recording, storing and serving data.

The system is characterized in that a user connected to the portal server and utilizing the configuration software may obtain data about a site and about an associated interactive form, the data is ultimately made available along with user data

to the generation server which writes the job order for execution by the navigation server. Form-submitted data accepted by a host is returned by the navigation server to the data repository and recorded for subsequent access to the site.

In an alternate embodiment, the system comprises virtual server *functions* characterized by the separate software functions of the application and is contained in one powerful server, which also servers as a portal server interface. In a preferred application, the system is distributed as separate software and hardware components described above. In this application, the separate servers are, in some cases, inter-connected by a separate and dedicated data network for achieving faster communication between the servers.

In still another aspect of the present invention a method for populating and submitting an interactive form by proxy is provided. The method comprises the steps of (a) obtaining data specific to a form, a site, and a requesting user, (b) writing a job order for navigating to the site for the purpose of populating and submitting the form to a host associated with the site, and (c) executing the job order.

In preferred embodiments of the method, the site is an electronic information page hosted on the Internet and the form is an electronic form located within the information page. Also in a preferred embodiment, the method includes a sub-execution routine for obtaining data from the site after successful form submission to the host sponsoring the site. For the first time, a user interacting with a portal service may register to WEB-service sites by proxy without having to physically fill out any interactive forms. Moreover, the method and apparatus of the present invention enables automated proxy services regarding a host of other WEB actions normally accomplished through physically populating an on-line form.

Brief Description of the Drawing Figures

Fig. 1 is an overview of an Internet portal system and network according to an embodiment of the present invention.

Fig. 2 is an exemplary plan view of a personalized Portal home page application as it may be seen on a display monitor according to an embodiment of the present invention.

Fig. 3 is a flow diagram illustrating user interaction with the Internet portal of fig. 1.

Fig. 4 is a block diagram illustrating a summarization software agent and capabilities thereof according to an embodiment of the present invention.

Fig. 5 is a logical flow chart illustrating an exemplary summarization process performed by the software agent of Fig. 4 operating in a user-defined mode.

Fig. 6 is a logical flow chart illustrating an exemplary summarization process performed by the software agent of Fig. 4 in a User-independent smart mode with minimum user input.

Fig. 7 is a block diagram illustrating a user-interactive template generation application according to an embodiment of the present invention.

Fig. 8 is a block diagram illustrating intelligent parsing capability according to an embodiment of the present invention.

Fig. 9 is a flow chart illustrating a semi-structured data retrieval process according to an embodiment of the present invention.

Fig. 10 is an overview of a communications network enhanced with automatic site registration according to an embodiment of the present invention.

Fig. 11 is a block diagram illustrating software architecture of an auto-registration suite according to an embodiment of the present invention.

Fig. 12 is a process flow diagram illustrating various process steps for auto-registration to a site and receiving data from the site according to an embodiment of the present invention.

Description of the Preferred Embodiments

According to a preferred embodiment of the present invention, a unique Internet portal is provided and adapted to provide unique services to users who have

obtained access via an Internet or other network connection from an Internet-capable appliance. Such an interface provides users with a method for storing many personal WEB pages and further provides search function and certain task-performing functions. The methods and apparatus of the present invention are taught in enabling detail below.

Fig. 1 is an overview of an Internet portal system 11 and Internet network 13 according to an embodiment of the present invention. Portal system 11, in this embodiment, operates as an ISP in addition to a unique network portal, but may, in other embodiments be implemented as a stand-alone Internet server. In yet other embodiments the service and apparatus described herein may also be provided by such as a search and listing service (AltaVista™, Yahoo™) or by any other enterprise hosting a WEB-connected server.

Internet 13 is representative of a preferred use of the present invention, but should not be considered limiting, as the invention could apply in other networks and combinations of networks.

ISP 15 in this embodiment comprises a server 31, a modem bank 33, represented here by a single modem, and a mass storage repository 29 for storing digital data. The modem bank is a convenience, as connection to the server could be by another type of network link. ISP 15, as is typical in the art, provides Internet access services for individual subscribers. In addition to well-known Internet access services, ISP 15 also provides a unique subscription service as an Internet portal for the purpose of storing many WEB pages or destinations along with any passwords and or personal codes associated with those pages, in a manner described in more detail below. This unique portal service is provided by execution of Portal Software 35, which is termed by the inventors the Password-All suite. The software of the invention is referred to herein both as the Portal Software, and as the Password-all software suite. Also, in much of the description below, the apparatus of the invention is referred to by the Password-All terminology, such as *the Password-All Server or Password-All Portal.*

ISP 15 is connected to Internet 13 as shown. Other equipment known in the art to be present and connected to a network such as Internet 13, for example, IP data

routers, data switches, gateway routers, and the like, are not illustrated here but may be assumed to be present. Access to ISP 15 is through a connection-oriented telephone system as is known in the art, or through any other Internet/WEB access connection, such as through a cable modem, special network connection (e.g. T1), ISDN, and so forth. Such connection is illustrated via access line 19 from Internet appliance 17 through modem bank 33.

In a preferred embodiment a user has access to Internet Password-All Portal services by a user name and password as is well known in the art, which provides an individualized WEB page to the subscriber. In another embodiment wherein a user has other individuals that use his or her Internet account, then an additional password or code unique to the user may be required before access to portal 31 is granted. Such personalized Portal WEB pages may be stored in repository 29, which may be any convenient form of mass storage.

Three Internet servers 23, 25, and 27, are shown in Internet 13, and represent Internet servers hosted by various enterprises and subscribed to by a user operating appliance 17. For example, server 23 may be a bank server wherein interactive on-line banking and account managing may be performed. Server 25 may be an investment server wherein investment accounts may be created and managed. Server 27 may be an airline or travel server wherein flights may be booked, tickets may be purchased, and so on. In this example, all three servers are secure servers requiring user ID and password for access, but the invention is not necessarily limited to just secure services.

In a preferred embodiment of the present invention, a subscribing user operating an Internet-capable appliance, such as appliance 17, connects to Password-All Portal system 11 hosted by ISP 15, and thereby gains access to a personalized, interactive WEB page, which in turn provides access to any one of a number of servers on Internet 13 such as servers 23, 25, and 27, without being required to enter additional passwords or codes. In a preferred embodiment the software that enables this service is termed Password-All by the inventors. Password-All may be considered to be a software suite executing on the unique server, and in some instances also on the user's station (client). Additional interactivity provided by portal

software 35 allows a connected user to search his listed pages for information associated with keywords, text strings, or the like, and allows a user to program user-defined tasks involving access and interaction with one or more Internet-connected servers such as servers 23, 25, and 27 according to a pre-defined time schedule. These functions are taught in enabling detail below.

Fig. 2 is an illustration of a personalized portal page as may be seen on a display monitor according to an embodiment of the present invention, provided by Password-All Portal software 35 executing on server 31, in response to secure access by a subscriber. Page 32 presents an interactive listing 34 of user-subscribed or member WEB pages, identified in this example by URL, but which may also be identified by any convenient pseudonym, preferably descriptive, along with user name and typically encrypted password information for each page. Listed in a first column under *destination*, are exemplary destinations LBC.com, My Bank.com, My Stocks.com, My shopping.com, Mortgage.com, and Airline.com. These are but a few of many exemplary destinations that may be present and listed as such on page 33. In order to view additional listings listed but not immediately viewable from within application 33, a scroll bar 35 is provided and adapted to allow a user to scroll up or down the list to enable viewing as is known in the art.

Items listed in list 34 in this example may be considered destinations on such as servers 23, 25, and 27 of Fig. 1. Typically the URL associated with an item on this list will not take a user to a server, per se, but to a page stored on a server. User names and password data associated with each item in list 34 are illustrated in respective columns labeled *user name*, and *password,* to the right of the column labeled *destination*. Each listing, or at least a portion of each listing, is a hyperlink invoking, when selected, the URL to that destination. In some instances a particular service may have more than one associated URL. For example, My Bank.com may have more than one URL associated for such as different accounts or businesses associated also with a single subscriber. In this case there may be a sub-listing for different destinations associated with a single higher-level listing. This expedient is not shown, but given this teaching the mechanism will be apparent to those with skill in the art.

BNSDOCID: <WO_____0180067A1 I >

In some embodiments one page 33 may be shared by more than one user, such as a husband and wife sharing a common account and subscription. An instance of this is illustrated herein with respect to the server labeled Mortgage.com wherein both a John and a Jane Doe are listed together under the column labeled user name. In another embodiment, a network of individuals, perhaps business owners, authorized co-workers, investment parties, or the like may share one application. In this way, system 11 may be adapted for private individuals as well as business uses.

After gaining access to application 33 which is served via Internet portal server 31 of Fig. 1, a user may scroll, highlight, and select any URL in his or her list 34 for the purpose of navigation to that particular destination for further interaction. Application 33 already has each password and user name listed for each URL. It is not necessary, however, that the password and user name be displayed for a user or users. These may well be stored transparently in a user's profile, and invoked as needed as a user makes selections. Therefore, a user is spared the need of entering passwords and user names for any destinations enabled by list 34. Of course, each list 34 is built, configured and maintained by a subscribing user or users, and an editing facility is also provided wherein a user may edit and update listings, including changing URLs adding and deleting listings, and the like.

In another aspect of the invention new listings for a user's profile, such as a new passthrough to a bank or other enterprise page, may be added semi-automatically as follows: Typically, when a user opens a new account with an enterprise through interaction with a WEB page hosted by the enterprise, the user is required to provide certain information, which will typically include such as the user's ID, address, e-mail account, and so forth, and typically a new user name and password to access the account. In this process the user will be interacting with the enterprise's page from his/her browser. A Password-All plug-in is provided wherein, after entering the required information for the new enterprise, the user may activate a pre-determined signal (right click, key stroke, etc.), and the Password-All suite will then enter a new passthrough in the user's Password. All profile at the Password-All Portal server.

In a related method for new entries, the enterprise hosting the Password-All Portal may, by agreement with other enterprises, provide log-in and sign-up services

at the Password-All Portal, with most action transparent to the user. For example, there may be, at the Password-All Portal, a selectable browser list of cooperating enterprises, such as banks, security services, and the like, and a user having a Password-All Portal subscription and profile may select among such cooperating enterprises and open new accounts, which will simultaneously and automatically be added to the Password-All Portal page for the user and to the server hosted by the cooperating enterprise. There may be some interactivity required for different accounts, but in the main, much information from the user's profile may be used directly without being re-entered.

The inventors have anticipated that many potential users may well be suspicious of providing passwords and user names to an enterprise hosting a Password-All Portal Server executing a service like Password-All according to embodiments of the present invention. To accommodate this problem, in preferred embodiments, it is not necessary that the user provide the cleartext password to Password. All. Instead, an encrypted version of each password is provided. When a user links to his passthrough page in Password-All at the Password-All Portal server, when he/she invokes a hyperlink, the encrypted password is returned to the user's system, which then, by virtue of the kept encryption key or master password, invokes the true and necessary password for connection to the selected destination. It is thus not necessary that cleartext passwords be stored at the Password-All Portal server, where they may be vulnerable to attack from outside sources, or to perceived misuse in other ways as well.

In a related safety measure, in a preferred embodiment of the invention, a user's complete profile is never stored on a single server, but is distributed over two or more, preferably more, servers, so any problem with any one server will minimize the overall effect for any particular user.

Password-All, as described above, allows a user to access a complete list of the user's usual cyberspace destinations, complete with necessary log-on data, stored in an encrypted fashion, so a user may simply select a destination (a hyperlink) in the Password-All list, and the user's browser then invokes the URL for the selected destination. In an added feature, Password-All may display banner ads and other

types of advertisement during the navigation time between a hyperlink being invoked and the time the destination WEB page is displayed.

In yet another embodiment of the invention, a user/subscriber need not access the Password-All page to enjoy the advantages of the unique features provided. In this variation, a Plug-In is provided for the subscriber's WEB browser. If the subscriber navigates by use of the local browser to a WEB page requiring a secure log-in, such as his/her on-line banking destination, when the subscriber is presented with an input window for ID and Password, the plug in may be activated by a predetermined user input, such as a hot key or right click of the mouse device. The plug-in then accesses, transparently, the Password-All page (which may be cached at the client), and automatically accesses and provides the needed data for log-on.

In yet another aspect of the invention a search option 37 allows a user to search list 34 for specific URLs based on typed input such as keywords or the like. In some cases, the number of URLs stored in list 34 can be extensive making a search function such as function 37 an attractive option. A criteria dialog box 51 illustrated as logically separated from and below list 34 is provided and adapted to accept input for search option 37 as is known in the art. In one embodiment, search option 37 may bring up a second window wherein a dialog box such as box 51 could be located.

In another aspect of the invention the search function may also be configured in a window invoked from window 33, and caused to search all or selected ones of listed destinations, and to return results in a manner that may be, at least to some extent, configured by a user. For example, a dialog box may be presented wherein a user may enter a search criteria, and select among all of the listed destinations. The search will then be access each of the selected destinations in turn, and the result may be presented to the user as each instance of the criteria is found, or results may be listed in a manner to be accessed after the search.

Preferably the search function is a part of the Password-All Portal software, available for all users, and may be accessed by hyperlinks in user's personal pages. In some embodiments users may create highly individualized search functions that may be stored in a manner to be usable only by the user who creates such a function.

In many aspects of the present invention, knowledge of specific WEB pages, and certain types of WEB pages, is highly desirable. In many embodiments characteristics of destination WEB pages are researched by persons (facilitators) maintaining and enhancing Password-All Portal software 35, and many characteristics may be provided in configuration modules for users to accomplish specific tasks. In most cases these characteristics are invoked and incorporated transparent to the user.

In yet another aspect of the present invention, the Password-All suite is structured to provide periodic reports to a user, in a manner to be structured and timed by the user, through the user's profile. For example, reports of changes in account balances in bank accounts, stock purchases, stock values, total airline travel purchases, frequent-flier miles, and the like may be summarized and provided to the users in many different ways. Because the Password-All Portal server with the Password-All software site handles a broad variety of transactional traffic for a user, there is an opportunity to summarize and collect and process statistics in many useful ways. In preferred embodiments of the invention such reports may be furnished and implemented in a number of different ways, including being displayed on the user's secure personal WEB page on the Password-All Portal.

In addition to the ability of performing tasks as described above, task results including reports, and hard documents such as airline tickets may be sent over the Internet or other data packet-networks to user-defined destinations such as fax machines, connected computer nodes, e-mail servers, and other Internet-connected appliances. All tasks may be set-up and caused to run according to user-defined schedules while the user is doing something else or is otherwise not engaged with the scheduled task.

In another embodiment of the present invention, recognizing the increasing use of the Internet for fiscal transactions, such as purchasing goods and services, a facility is provided in a user's profile to automatically track transactions made at various destinations, and to authorize payment either on a transaction-by-transaction basis, or after a session, using access to the user's bank accounts, all of which may be pre-programmed and authorized by the user.

Other functions or options illustrated as part of application 35 include a last URL option 41, an update function 43, and an add function 45. Function 41 allows a user to immediately navigate to a last visited URL. Update function 43 provides a means of updating URLs for content and new address. An add function enables a user to add additional URLs to list 34. Similarly, function 45 may also provide a means to delete entries. Other ways to add accounts are described above. It should be noted that the services provided by the unique Password-All Portal in embodiments of the present invention, and by the Password-All software suite are not limited to destinations requiring passwords and user names. The Password-All Portal and software in many embodiments may also be used to manage all of a user's bookmarks, including editing of bookmarks and the like. In this aspect, bookmarks will typically be presented in indexed, grouped, and hierarchical ways.

There are editing features provided with Password-All for adding, acquiring, deleting, and otherwise managing bookmarks. As a convenience, in many embodiments of the invention, bookmarks may be downloaded from a user's Password-All site, and loaded onto the same user's local browser. In this manner, additions and improvements in the bookmark set for a user may be used without the necessity of going to Password-All. Further, bookmarks may be uploaded from a user's local PC to his/her home page on the Password-All site by use of one or more Password-All plug-ins.

It will be apparent to the skilled artisan, given the teaching herein, that the functionality provided in various embodiments of the invention is especially applicable to Internet-capable appliances that may be limited in input capability. For example, a set-top box in a WEB TV application may well be without a keyboard for entering IDs and Passwords and the like. In practice of the present invention keyboard entry is minimized or eliminated. The same comments apply to many other sorts of Internet appliances.

In preferred embodiments of the invention, once a subscriber-user is in Password-All, only an ability to point-and-click is needed for all navigation. To get into the Password-All site, using a limited apparatus, such as an appliance without a

keyboard or keypad, a Smartcard or embedded password may be used, or some other type of authentication.

It will be apparent to one with skill in the art that an interactive application such as application 33 may be provided in a form other than a WEB page without departing from the spirit and scope of the present invention. For example, an application such as application 33 may be provided as a downloadable module or program that may be set-up and configured off-line and made operational when on-line.

Fig. 3 is a flow diagram illustrating user interaction with the Internet Password-All Portal of fig. 1. The following process steps illustrated, according to an embodiment of the present invention, are intended to illustrate exemplary user-steps and automated software processes that may be initiated and invoked during interaction with an Internet portal of the present invention such as portal 31 of Fig. 1. In step 53 a user connects to the Internet or another previously described switched-packet network via a compatible appliance such as Internet appliance 17 of Fig. 1.

At step 55, a user enters a user-name and password, which, in one embodiment, may simply be his ISP user name and password. In another embodiment, a second password or code would be required to access an Internet portal such as portal server 31 of Fig. 1 after logging onto the Internet through the ISP. In some cases, having a special arrangement with the ISP, there may be one password for both Internet access through the ISP and for Password-All. At step 57 a personal WEB page such as page 32 of Fig. 2 is displayed via Internet portal server 31. At minimum, the personalized WEB page will contain all user configured URLs, and may also be enhanced by a search function, among other possibilities.

In step 58 a user will, minimally, select a URL from his or her bookmarked destinations, and as is known by hyperlink technology, the transparent URL will be invoked, and the user will navigate to that destination for the purpose of normal user interaction. In this action, the Password-All Portal software transparently logs the user on to the destination page, if such log-on is needed.

At step 60 the user invokes a search engine by clicking on an option such as described option 37 of Fig. 2. At step 62, the user inputs search parameters into a

provided text field such as text field 51 of Fig. 2. After inputting such parameters, the user starts the search by a button such as button 52. The search engine extracts information in step 64. Such information may be, in one option, of the form of URLs fitting the description provided by search parameters. A searched list of URLs may be presented in a separate generated page in step 66 after which a user may select which URL to navigate to. In an optional search function, the user may provide search criteria, and search any or all of the possible destinations for the criteria.

In another embodiment wherein WEB pages are cached in their presentable form, information extracted in step 64 may include any information contained in any of the stored pages such as text, pictures, interactive content, or the like. In this case, one displayed result page may provide generated links to search results that include the URL associated with the results. Perhaps by clicking on a text or graphic result, the associated WEB page will be displayed for the user with the result highlighted and in view with regards to the display window.

Enhanced Agent for WEB Summaries

In another aspect of the present invention, a software agent, termed a gatherer by the inventors, is adapted to gather and return summary information about URLs according to user request or enterprise discretion. This is accomplished in embodiments of the present invention by a unique scripting and language parsing method provided by the inventor wherein human knowledge workers associated with the service provide written scripts to such a gatherer according to subscriber or enterprise directives. Such a software gatherer, and capabilities thereof, is described in enabling detail below.

Referring now to Fig. 1, there is illustrated an exemplary architecture representing a portal service-network which, in this case is hosted by ISP 15. Portal software 35 in this embodiment executes on portal server 31 set-up at the ISP location. Mass repository 29 is used for storing subscriber information such as passwords, login names, and the like. Internet servers 23, 25, and 27 represent servers

that are adapted to serve WEB pages of enterprises patronized by a subscriber to the portal service such as one operating Internet appliance 17.

The main purpose of portal software 35 as described above with reference to Fig. 2, is to provide an interactive application that lists all of the subscriber's WEB sites in the form of hyperlinks. When a user invokes a hyperlink from his personal list, software 35 uses the subscriber's personal information to provide an automatic and transparent login function for the subscriber while jumping the subscriber to the subject destination.

Referring again to Fig. 2, an interactive list 34 containing user-entered hyperlinks and a set of interactive tools is displayed to a subscriber by portal software 35 of Fig. 1. One of the tools available to a subscriber interacting with list 34 is agent (software) 39. Agent 39 may be programmed to perform certain tasks such as obtaining account information, executing simple transactions, returning user-requested notification information about upcoming events, and so on. Search function 37 and update function 43 may be integrated with agent 39 as required to aid in functionality.

It is described in the above disclosure that agent 39 may, in some embodiments, search for and return certain summary information contained on user-subscribed WEB pages, such as account summaries, order tracking information and certain other information according to user-defined parameters. This feature may be programmed by a user to work on a periodic time schedule, or on demand.

In the following disclosure, enhancements are provided to agent 39. Such enhancements, described in detail below, may be integrated into agent 39 of portal software 35 (Fig.'s 1 and 2); and may be provided as a separate agent or gatherer to run with portal software 35; or may, in some embodiments, be provided as a standalone service that is separate from portal software 35.

Fig. 4 is a block diagram illustrating a summarization software agent 67 and various capabilities and layers thereof according to an embodiment of the present invention. Summarization agent 67, hereinafter termed gatherer 67, is a programmable and interactive software application adapted to run on a network server. Gatherer 67 may, in one embodiment, be integrated with portal software 35 of

Fig. 1 and be provided in the form of a software module separate from agent 39 (Fig. 2). In another embodiment, gatherer 67 may be a part of agent 39 as an enhancement to the function of that agent as previously described. In still another embodiment, gatherer 67 may be provided as a parent or client-side application controlled by a separate service from the portal service described above.

In this exemplary embodiment gatherer 67 is a multi-featured software application having a variety of sub-modules and interface modules incorporated therein to provide enhanced function. Gatherer 67 has a client/service interface layer 69 adapted to enable directive input from both a client (user) and a knowledge worker or workers associated with the service. A browser interface 77 is provided in layer 69, and adapted to provide access to application 67 from a browser running on a client's PC or other Internet or network appliance. Interface 77 facilitates bi-directional communication with a user's browser application (not shown) for the purpose of allowing the user to input summary requests into gatherer 67 and receive summary results. Interface 77 supports all existing network communication protocols such as may be known in the art, and may be adapted to support future protocols.

Layer 69 also comprises a unique input scripting module 79 that is adapted to allow a human knowledge worker to create and supply directive scripts containing the site logic needed by gatherer 67 to find and retrieve data from a WEB site. In this case, gatherer 67 executes and runs on a network server such as server 31 of Fig. 1. However, this is not required in order to practice the present invention.

It is assumed in this example that gatherer 67 is part of the portal software suite 35 running on server 31 of Fig. 1. Gatherer 67 may be provided as several dedicated agents, or as one multi-functional agent without departing from the spirit and scope of the present invention. For example, one gatherer 67 may be scripted and programmed to execute a single user request with additional gatherers 67 called upon to perform additional user-requests. Alternatively, one gatherer 67 may be dedicated and assigned to each individual user and adapted to handle all requests from that user.

Interface layer 69 facilitates exchange of information from both a client and a knowledge worker. A client operating a WEB browser with an appropriate plug-in is enabled to communicate and interact with gatherer 67. For example, a user may enter

a request to return a summary of pricing for all apartments renting for under $1000.00 per month located in a given area (defined by the user) from apartments.com (one of user's registered WEB sites). The just mentioned request would be categorized as either a periodic request, or a one time (on demand) request. The communicated request initiates a service action wherein a knowledge worker associated with the service uses module 79 to set-up gatherer 67 to perform it's function. Module 79 is typically executed from a network-connected PC operated by the knowledge worker.

According to an embodiment of the present invention, a unique scripting method facilitated by module 79 is provided to enable gatherer 67 to obtain the goal information requested by a user. For example, the above mentioned example of WEB-site apartments.com has a specific HTML (hyper-text-markup-language) logic that it uses to create its site and post its information. Such site logic is relatively standard fare for a majority of different sites hosted by different entities. Using this knowledge, a knowledge worker creates a site-specific script or template for gatherer 67 to follow. Such a template contains descriptions and locations of the appropriate fields used, for example, at apartments.com. Apartment description, location, deposit information, rental information, agent contact information, and other related fields are matched in terms of location and label description on the template created with module 79. Completed templates are stored in a database contained in a storage facility such as, perhaps, repository 29 of Fig. 1. Such templates may be reused and may be updated (edited) with new data.

In one embodiment, one script may contain site logics for a plurality of WEB pages, and instructions for specific navigational instruction and password or login information may be contained therein and executed serially, such as one site at a time. It is important to note that the knowledge worker or workers may perform much of their scripting via automatic controls such as by object linking and embedding (OLE) and a minor portion of scripting may be performed manually in an appropriate computer language, many of which are known in the art).

Gatherer 67 also has a process layer 71 adapted for internal information gathering and parameter configuration. An optional portal server interface 81 is provided and adapted to allow gather 67 to provide updated information to a user's list

of hyperlinks and also to obtain data from portal server 31 if required. For example, required hyperlinks may be mirrored from a user's home page to a scripting template for navigational purposes. In an embodiment wherein gatherer 67 is part of a standalone service, a convention for providing user login information may be supplied at the client's end when a request is made. For example, an encrypted password may be supplied by a client plug-in and gatherer 67 may temporarily borrow the user's encryption key when auto login is performed.

An appliance configuration module 83 is provided and adapted to allow a user to define and configure an Internet appliance to communicate with the service and receive summary information. Such appliances may include but are not limited to palm top PC's, lap top PC's, cellular telephones, WEB TV's, and so on. Typically, a user will be presented a configuration WEB page from a network server that displays in his browser window on his desktop PC. The page contains an interface for communicating device parameters and communication protocol types to module 83. In this way, a user may configure a preferred device for receipt of summary information. Device parameters and communication protocols inherent to such a device are incorporated into the scripting of the site template and are used as instructions for WEB summary delivery.

A navigation layer 73 is provided and adapted to perform the function of external site navigation and data gathering for gatherer 67. To this end, a communication interface/browser control module 85 is provided and adapted to function as a WEB browser to access WEB sites containing WEB data. Control 85 receives it's instruction from the scripted template created by the knowledge worker.

A parsing engine 87 is provided and adapted to parse individual WEB sites according to a template created via scripting module 79. Parsing engine 87 may be a Pearl engine, an IE HTML engine, or any other or combination of known parsing engines. The template (not shown) tells control 85 and parsing engine 87 where to go and what fields at the destination site to look for to access desired data. Once the data fields are located, parsing engine 87 gathers current data in the appropriate field, and returns that data to the service for further processing such as data conversion, compression and storage, and the like.

Because WEB sites use tools that use consistent logic in setting up their sites, this logic may be used by the summarization service to instruct control 83 and parsing engine 87. The inventor provides herein an exemplary script logic for navigating to and garnishing data from amazon™.com. The hyperlinks and/or actual URLs required for navigation are not shown, but may be assumed to be included in the template script. In this example, a company name Yodlee (known to the inventors) is used in the script for naming object holders and object containers, which are in this case Active X™ conventions. In another embodiment, Java™ script or another object linking control may be used. The scripted template logic example is as follows:

```
# Site amazon.orders.x - shows status of orders from Amazon

login( 7 );

get( "/exec/obidos/order-list/" );

my @tables = get_tables_containing_text( "Orders:" );

my $order_list = new Yodlee::ObjectHolder( 'orders' );
$order_list->source( 'amazon' );
$order_list->link_info( get_link_info() );

my @href_list;
my @container_list;

foreach my $table ( @tables ) {
        my @rows = get_table_rows();

foreach my $i ( 0 .. $#rows ) {
        select_row( $i );
        my $text = get_text( $rows[ $i ] );
        next if $text =~ /Orders:|Status/;

        my @items = get_row_items();
        next unless @items >= 4;

        my( $order_num, $date, $status );
        select_cell( 1 );
        $order_num = get_cell_text();

        my $href = get_url_of_first_href( get_cell() );

        select_cell( 2 );
```

```
            $date = get_cell_text();
            select_cell( 3 );
            $status = get_cell_text();

            next unless defined $order_num and defined $date and defined $status;

            my $order = new Yodlee::Container( 'orders' );
            $order->order_number( $order_num );
            $order->date( $date );
            $order->status( $status );

            $order_list->push_object( $order );

            if( defined $href ) {
              push( @href_list, $href );
              push( @container_list, $order );
    foreach my $i ( 0 .. $#href_list ) {

            get( $href_list[ $i ] );

            @tables = get_tables_containing_text( "Items Ordered:" );

     foreach my $table ( @tables ) {
            my @rows = get_table_rows();

     foreach my $j ( 0 .. $#rows ) {
            select_row( $j );

            my $href = get_url_of_first_href( get_row() );

            next unless defined $href;

            my @child_list = get_children( get_row(), 'a' );
            next unless defined $child_list[ 0 ];

            my $text = get_text( $child_list[ 0 ] );

            $container_list[ $i ]->description( $text );
        }
      }
    }

    result( $order_list );
```

The above example is a script that instructs control 85 and parser 87 to navigate to and obtain data from Amazon™.com, specifically that data that reflects the user's current order status. Scripts may also be written to obtain virtually any type of text information available from any site. For example, a user may wish to obtain the New York Times headlines, the top ten performing stocks, a comparative list of flights from San Francisco to New York, etc. In one embodiment, metadata may be associated with and used in-place of the actual scripted language for the purpose of reducing complication in the case of many scripts on one template.

A data processing layer 75 is provided and adapted to store, process, and present returned data to users according to enterprise rules and client direction. A database interface module 89 is provided and adapted to provide access for gatherer 67 to a mass repository such as repository 29 of Fig. 1, for the purpose of storing and retrieving summary data, templates, presentation directives, and so on. Gatherer agent 67 may also access data through interface 89 such as profile information, user account and URL information, stored site logics and so on. Data scanned from the WEB is stored in a canonical format in a database such as repository 29, or in another connected storage facility. All stored data is, of course, associated with an individual who requested it, or for whom the data is made available according to enterprise discretion.

A summarization page module 91 is provided and adapted to organize and serve a WEB summary page to a user. Module 91, in some embodiments, may immediately push a WEB summary to a user, or module 91 may store such summarized pages for a user to access via a pull method, in which case a notification may be sent to the user alerting him of the summary page availability. Summarization module 91 includes an HTML renderer that is able to format data into HTML format for WEB page display. In this way, e-mail messages and the like may be presented as HTML text on a user's summarization page. Moreover, any summary data from any site may include an embedded hyperlink to that site. In this way, a user looking at an e-mail text in HTML may click on it and launch the appropriate e-mail program. Other sites will, by default, be linked through the summary page.

Many users will access their summary data through a WEB page as described above, however, this is not required in order to practice the present invention. In some embodiments, users will want their summary information formatted and delivered to one of a variety of Internet-capable appliances such as a palm top or, perhaps a cell phone. To this end, the renderer is capable of formatting and presenting the summary data into a number of formats specific to alternative devices. Examples of different known formats include, but are not limited to XML, plain text, VoxML, HDML, audio, video, and so on.

In a preferred embodiment of the present invention, gather 67 is flexible in such a way as it may act according to enterprise rules, client directives, or a combination of the two. For example, if a user makes a request for summary data about a user/subscribed WEB page to be periodically executed and presented in the form of a HTML document, then gather 67 would automatically access and analyze the required internal information and user provided information to formulate a directive. Using scripting module 79, a knowledge worker provides a template (if one is not already created for that site) that contains the "where to go" and "what to get" information according to site logic, user input, and known information.

Alternatively, if a user requests a summary about data on one of his sites such as, perhaps, current interest rates and re-finance costs at his mortgage site, the service may at it's own discretion provide an additional unsolicited summary from an alternate mortgage site for comparison. This type of summarization would be designed to enhance a user's position based on his profile information. In this case, updated data about latest interest rates, stock performances, car prices, airline ticket discounts, and so on would be stored by the service for comparative purposes. If a user request for a summary can be equaled or bettered in terms of any advantage to the user, such summary data may be included.

In many cases, created templates may be re-used unless a WEB site changes it's site logic parameters, in which case, the new logic must be accessed and any existing templates must be updated, or a new template may be created for the site. The templates contain site-specific script obtained from the site and stored by the knowledge workers. In one embodiment, companies hosting WEB pages

automatically provide their site logics and any logic updates to the service by virtue of an agreement between the service and the WEB hosts.

In an alternative embodiment gatherer 67 may be implemented as a client application installed on a user's PC. In this embodiment, a user would not be required to supply log-in or password codes. Summarization scripts may be sent to the client software and templates may be automatically created with the appropriate scripts using log-in and password information encrypted and stored locally on the user's machine.

In addition to providing WEB summary information, gatherer 67 may also be used to provide such as automatic registration to new sites, and for updating old registration information to existing sites. For example, if a user whishes to subscribe, or register at a new site, only the identification of the site is required from the user as long as his pertinent information has not changed. If a new password or the like is required, gatherer 67 through control module 73 may present login or password codes from a list of alternative codes provided by a user. In another embodiment, a database (not shown) containing a wealth of password options may be accessed by gatherer 67 for the purpose of trying different passwords until one is accepted by the site. Once a password or log-in code is accepted, it may be sent to a user and stored in his password list and at the network level.

It will be apparent to one with skill in the art that a software application such as gatherer 67 may be implemented in many separate locations connected in a data network. For example, a plurality of gatherer applications may be distributed over many separate servers linked to one or more mass repositories. Client applications include but are not limited to a WEB-browser plug-in for communicating to the service. Plug-in extensions may also be afforded to proxy servers so that auto-login and data access may still be performed transparent to a user.

In another embodiment, plug-ins enabling communication with gatherer 67 may be provided and configured to run on other network devices for the purpose of enabling such a device to initiate a request and get a response without the need for a desktop computer.

In most embodiments a user operating a desktop PC will order a one time or periodic summary related to some or all of his subscribed WEB sites. A logical flow of an exemplary request/response interaction is provided below.

Fig. 5 is a logical flow chart illustrating an exemplary summarization process performed by the software agent of Fig. 4 operating in a user-defined mode. In step 93, a user has initiated a new request for a summary (summary order). It is assumed for the purpose of discussion, that the request of step 93 involves a site wherein no template has been created. In step 95, the request is received and analyzed. A knowledge worker will likely perform this step. The new request may be posted to the user's portal home page, sent directly to gatherer 67, or even communicated through e-mail or other media to the service.

In step 97 a knowledge worker accesses particular site logic associated with the request URLs. For example, if the request involves a plurality of URLs, then all site logics for those URLs are accessed. Logic may be available in a repository such as repository 29 of Fig. 1 if they were obtained at the time of user registration to a particular URL, or sent in by WEB-site hosts shortly after registration. If it is a completely new URL, then the logic must be obtained from the site. In most cases however, the logic will be known by virtue of a plurality of users accessing common URLs. Therefore cross-linking in a database of logic/user associations may be performed to access a logic for a site that is new to one particular user, but not new to another.

In step 99, the knowledge worker creates a template by virtue of scripting module 79 (Fig. 4) containing all site logic, URLs, log-in and password information, and the user request information. As described previously, templates may be re-used for a same request. In most cases, scripting may be mostly automated with minimum manual input performed by the knowledge worker. In many cases, an existing template will match a new request exactly, and may be re-used. In that case steps 97, 99, and 101 would not be required.

In step 101 the template is stored and associated with the requesting user. The stored template may now be retrieved at a scheduled time for performing the summary gathering. At step 103, a browser control such as module 85 of Fig. 4 is activated to

access the stored template and navigate to specified URLs for the purpose of gathering summary data. If a timing function is attributed to the template stored in step 101, then the template may self execute and call up the browser function. In another embodiment, the knowledge worker may notify the browser control to get the template for it's next task. In some embodiments, a plurality of controls may be used with one template as previously described.

In step 105, automatic log-in is performed, if required, to gain access to each specified URL. In step 107, a specified WEB page is navigated to and parsed for requested data according to the logic on the template. If there are a plurality of WEB –pages to parse, then this step is repeated for the number of pages. A variety of parsing engines may be used for this process such as an IE™ parser, or a Pearl™ parser. Only the requested data is kept in step 107.

A request may be an on-demand request requiring immediate return, or a scheduled request wherein data may be posted. At step 109, such logic is confirmed. If the data is to be presented according to a periodic schedule, then summary data parsed in step 107 is stored for latter use in step 111. In step 113, the summary data is rendered as HTML if not already formatted, and displayed in the form of a summary WEB page in step 115. The summary page may be posted for access by a user at a time convenient to the user (pull), or may be pushed as a WEB page to the user and be made to automatically display on the user's PC. Notification of summary page availability may also be sent to a user to alert him of completion of order.

If the summary data is from a one-time on-demand request and required immediately by a user, then a network appliance and data delivery method (configured by the user) is confirmed, and the data is rendered in the appropriate format for delivery and display in step 117. In step 119, the summary data is delivered according to protocol to a user's designated appliance. In step 121 a user receives requested information in the appropriate format.

It will be apparent to one with skill in the art that there may be more or fewer logical steps as well as added sub-steps than are illustrated in this example. For example, step 105 may in other embodiments include sub-steps such as getting an encryption key from a user. In still another embodiment, part of a request may be

rendered as HTML as in step 113 while certain other portions of the same request data might be rendered in another format and delivered via alternative methods. There are many possibilities.

The method and apparatus of the present invention may be used to present summaries to users without user input. Process logic such as this is detailed below.

Fig. 6 is a logical flow chart illustrating an exemplary summarization process performed by the software agent of Fig. 4 in a User-independent smart mode with minimum or no user input. In step 117 an enterprise-initiated summary process begins. In this case, the enterprise may be assisting a user in finding a better deal or, perhaps presenting the individual with summaries from and links to alternative pages not yet subscribed to by a user.

In step 119, a database containing user information and parameters is accessed and reviewed. Certain information specific to a user may be required to initiate an enterprise-sponsored summary report. At step 121, the knowledge worker accesses the site logic specific to the specified target site or sites for summarization. In step 123, the knowledge worker modifies an existing user template, or creates a new one if necessary. At step 125 the template is stored in a repository such as repository 29 and associated with the user.

As described in Fig. 5, the template either self-executes according to a timed function and invokes a browser control such as control 85 (Fig. 4), or is accessed by control 85 as a result of task notification. In step 127, the browser control begins navigation. Auto logins are performed, if required, in step 129 to gain access to selected sites. If the WEB pages are new to a user, and the user has no registration with the WEB site, then through agreement, or other convention, the service may be provided access to such sites. Such an agreement may be made, for example, if the host of the WEB site realizes a possibility of gaining a new customer if the customer likes the summary information presented. In many other situations, no password or login information is required to obtain general information that is not personal to a client.

In step 131, all sites are parsed for summary data and stored in canonical fashion in step 133. At step 135, the data is compiled and rendered as HTML for presentation on a summary page. In step 137, a WEB summary containing all of the data is made available to a user and the user is notified of its existence.

Providing certain information not requested by a user may aid in enhancing a user's organization of is current business on the WEB. Moreover, unsolicited WEB summaries may provide better opportunities than the current options in the user's profile. Of course, assisting a user in this manner will require that the enterprise (service) have access to the user's profile and existing account and service information with various WEB sites on the user's list. A user may forbid use of a user's personal information, in which case, no enterprise-initiated summaries would be performed unless they are conducted strictly in an offer mode instead of a comparative mode.

Enhanced Data Parsing and Logic Integration

In another aspect of the present invention, a software application is provided and adapted to recognize and parse data from semi-structured, WEB-based data sources on behalf of a user.

Referring to Figs. 1-3 of application S/N 09/208,740, a portal system for interacting with WEB-based data is enabled by portal software 35 running on portal server 31 located, in this case, at ISP 15 of Fig. 1. Software 35 allows users who are connected to server 31 via Internet network connection, to access pre-configured URLs without requiring manual log-in procedures to be performed.

Portal Interface 37 is an Interactive WEB page, which may be personalized to a specific user. In the example of Fig. 2, page 37 displays user-visited links and encrypted passwords to those links. Page 37 also provides software tools a user may invoke to search for data either from his or her stored hyperlinks, or from specified fields within the pages defined by the hyperlinks. The example process of Fig. 3 illustrates the above-described function.

Figs. 4-6 of S/N 09/323,598 illustrate a method and apparatus for performing meta-summarization of WEB-based data on behalf of a user according to user request. Navigation to search the data is performed by proxy using what is termed a gatherer agent (software). Site-logic scripts are prepared by knowledge workers for navigation to the separate user sites. The scripts are typically Java-based executable routines combined with specific data descriptions and, if required, field location information provided by the user at the time of request. A user may request summary data from a plurality of sites and get the data returned on demand, or have it held by the system for later access.

As previously described in the background section, field locations for data must be provided in order for the parsing engine to identify and return specified data in a structured arrangement. An input template provides instruction to the parser concerning what data to obtain for return. The template is created in part by user input through the portal interface and in part by knowledge worker providing data-navigation instruction, which may include field locations for structured data.

The system partly relies on the fact that data requested by a user will appear at a same location (field) within a WEB page every time the system searches for it. The field location then becomes a critical part of finding the data. The inventor provides an enhancement to the WEB-summary system that allows the system to search for and parse data without the constraint of adhering to rigid data structures. Such a method and apparatus is described below.

Fig. 7 is a block diagram illustrating a user-interactive set-up application 139 according to an embodiment of the present invention. Application 139 is, in a preferred embodiment, a graphical tool provided to users and accessible through their respective portal pages. A user invokes application 139 when he or she desires to pre-configure profiled data sources for meta-summarization. In a preferred embodiment, activation of application 139 causes an interactive browser window to open within a user's portal page. By working within the window, a user may pre-configure sites for data acquisition.

Application 139 comprises an initialization layer 141 adapted for user interface, and a document type definition (DTD) generation layer 143 adapted to

generate DTDs, which describe selected data sources. Initialization layer 141 has a browser control module 145 provided therein and adapted as a browser extension for navigation to selected URLs. Application 139 may be integrated with a user's browser application such that the presented window of application 139 is a browser window as described above. A user-data interface 147 is provided within layer 141 and adapted as a data-input field for entering URLs for navigation purposes. By entering a URL into interface 147, browser control module 145 causes navigation to and presentation of the WEB page defined by the entered URL.

Presentation of the selected WEB page is conducted utilizing the well-known multi-window technology such that the WEB page may be displayed within the window of application 139 while leaving other controls attributed to application 139 visible and accessible to a user. One such control is a block selection module 149, which is adapted as a user-controlled highlighter for selecting specific parts of a WEB page containing data that is desired by the user to be accessed. Module 149 may function by combination of keyboard manipulation and mouse click similar to well-known text selection tools available in word applications.

A selection-submission module 151 is provided within layer 141 and implemented to allow a user to submit selected portions of a WEB page to the portal system for DTD generation. Highlighting or blocking a section of a WEB page defines the boundaries for data search. A user must also specify the type of data to be extracted from the highlighted area of the WEB page.

In practice a user may call-up many WEB pages, highlight (define) sections of each page, and provide data descriptions for data extraction with regard to each page all in one session with application 139. The method described above comprises a meta-data request wherein the returned results are aggregated and displayed to, or otherwise made available to users according to enterprise rules. The output of layer 141 is a highlighted portion of a "structure tree" describing the HTML data structure used to display the data on the target WEB site along with associated data-type descriptions input by a user.

A data structure tree defines how data is displayed on a WEB page in HTML format. An application program interface (API), which is part of the normal browser

function understands the HTML data structure used in the construction of a WEB page, which is available at the source. Therefore, a specific structure tree is captured by browser module 145 every time an associated WEB page is displayed in application window 139. User-highlighted portions of a WEB page define "portions" of the overall data structure tree of that page. Therefore, the output of layer 141 comprises an entire data structure tree for a WEB page, a highlighted portion of that data structure, and a description of the type of human-readable data to be extracted from that portion of the tree.

Layer 143 is responsible for producing DTDs adapted to facilitate data search and extraction. A DTD generator is provided within layer 143 and adapted to generate a DTD of a data source using information supplied by layer 141. A DTD is a template written in a language such as XML that is understood by a browser/data-parsing application. In a preferred embodiment a DTD is created in Extensible Markup Language (XML), however, any standard script language may be employed such as Standard Generalized Markup Language (SGML). A DTD is generated for each portion of a WEB page a user highlights.

A database interface module 155 is provided within layer 143 and adapted to allow interface to a connected database for the purpose of obtaining site-logic scripts for navigation, which include log-in blocks for performing automated log-in function for sites requiring passwords. In some cases new site-logic scripts are required if data is to be extracted from a new page. Interface 155 may also be used to obtain any other useful information.

A DTD/site-logic integration module 157 is provided within layer 143 and adapted to create a routine combining site logic and DTD templates in such a way that navigation and data extraction is performed seamlessly and automatically. For example, a site logic script with a log-in block is used to navigate to and access a particular URL, then a DTD template portion reveals where on that URL page to look for data and what type of data to extract from the URL page. Information gathering for a next URL would follow a next site-logic and DTD instruction contained in the routine and marked for that URL and so on. An automated navigation and data

extraction routine may facilitate a plurality of URLs included in one data search execution.

It is conceivable that one routine containing the required instruction blocks could facilitate automated navigation and information retrieval from a great many URLs during one automated routine on behalf of a user. A single WEB site or a combination of, or all of a user's WEB sites may be processed for data during a single automated process. All that is required of a user is to first set-up a URL or URLs for data searching by navigating to them, highlighting portions contained therein, and providing a description of the type of data to be extracted. Once set-up, summary data from the affected URL pages may be periodically gathered on behalf of the user and presented in a variety of ways, some of which are described in disclosure pertaining to S/N 09/208,740 and S/N 09/323,598.

It will be apparent to one with skill in the art that application 139 as exemplified above is provided to execute as a network-supported application on a client. In this example, layer 141 is provided to execute as a client-side application while layer 143 is provided and executed at server side. Communication and cooperation between layers 141 and 143 is achieved through appropriate network-interface technologies known in the art. Proxy navigating and data gathering is executed at the server side on behalf of a user. In one embodiment all of the function of application 139 may be provided at the client's side. In another, all of the function may be applied on the server side, in which case a knowledge worker associated with the server may initialize the sites for search.

It will also be apparent to one with skill in the art that other modules responsible for added function may, if desired, be provided within application 139 without departing from the spirit and scope of the present invention. For example, a module for sectoring a pre-configured data search and scheduling each sector for execution may be provided within layer 141. In this case, a user may schedule alternate data searches of separate groups of URLs. Other lesser controls (not illustrated) may also be provided such as edit controls, clear selection, cancel search, and so on. There are many possibilities. The inventor intends that application 139

represent just one simple implementation out of many possible alternative implementations.

Fig. 8 is a block diagram illustrating intelligent navigation and parsing capability according to an embodiment of the present invention. A navigation/parsing application 159 is provided for the purpose of navigating to and parsing data from WEB-based data sources on behalf of a user. Application 159 may be considered somewhat analogous to the "gathering agent" described in S/N 09/323/598. Application 159 is responsible for navigating to selected URLs and extracting data from them for return to users. Enhanced capability provided in part by enhanced input data for navigation and in part by built-in intelligence enables application 159 to operate more intelligently and in a more streamlined manner with respect to workload.

Input to application 159 comprises DTD data and site-logic data interwoven into an instruction routine that provides instruction for navigation, log-in (if required), and data parsing. A directional arrow labeled DTD/Site-logic illustrates data input. A navigation and parsing sequence 161 is executed based on input data instruction. Parsing intelligence is enhanced by virtue of data reference libraries 163 and 165. Library 163 is responsible for enabling text matches and associations.

Library 163 is flexible such that a parsed word may be associated with a variety of similar words or phrases. For example, if data is being parsed from an investment site quoting stock prices, then the appearance of the word portfolio may be equated with not only the exact word, but also with a similar word or phrases known to equate with the word such as "stock portfolio", "current portfolio", "view portfolio", and so on. This enables a user to approximate the description of the type of data that is extracted from a highlighted portion of a WEB page. Moreover, if a WEB master changes the description of the data between searches, an intelligent parser will still be able to find the data through word and phrase association techniques.

Library 165 contains examples of style variations that may be used in the presentation of data on a WEB page. Library 165 enables the parser to understand data presented in a variety of styles such as italic, bold, underlined, small fonts, larger fonts, text formats, font styles, and so on. In this way, a parser may still find the

requested data even if a WEB master changes the style or format of presentation between data searches.

In one embodiment of the present invention, successful request histories may be compiled and used to enhance parsing and data return. Parsing is, in this example, restricted to a highlighted portion or portions of a WEB page, wherein the requested data presumably resides. Because of this, it is likely that only a few differing types of information will be available in the highlighted portion of a WEB page. Therefore, past request records may be used in an attempt to eliminate or confirm a data type residing in a highlighted section under consideration. This embodiment assumes that a current request for data does not register any matching words or phrases contained in the data type or in parsing intelligence.

To further illustrate, assume that there are two different data types available in a highlighted portion of a WEB page. If the page is a banking page, one of the data sets may be a list of available loans and current interest rates. The other data set included in the section may be a list of most recent savings account deposits and withdraws. By parsing all of the data in the section, and comparing the data to a previous (successful) routine, it may be determined which type of data was previously returned. By comparing the previous natural language request to the current one, it may be determined if the requests are similar in nature. If a previous request read "return available loan options and current rates" it can be determined that the loan data did match that request. If a current non-registering request reads "send me my balance", it may be determined that the current request does not register with the system, but it is also not analogous to the previous successful request. Therefore, the data set corresponding to the request for loan information is discarded and the remaining data is returned including the account information.

The above example serves to illustrate adaptive intelligence that is provided to navigation and parsing sequence 161. It is assumed herein that a user will only select portions of a WEB page where desired data resides. Therefore, it is likely in many cases that one or only a few types of data will be included in a selected portion. This fact makes it easier to determine what data will be returned to a user. It is also noted here that all of the data from a selected portion of a WEB page is retrieved from a

source. Parsing of the data after it is retrieved determines which data will be returned to a user. Data that does not fit a request for that section is discarded. In one embodiment of the present invention, all data from a section is returned by default if a request cannot be matched to specific data in the section.

In still another embodiment, a WEB master may change the location of data on a WEB page between routine data searches such that it is moved out of a user's highlighted section of the page. In this case, a routine could be performed to find the moved data if the WEB master has retained the data structure. To accomplish this, the system compares the old highlighted structure with the new structure of the WEB page and matches the structures. If the data contained in the new section matches the current request, then a correction is made.

Application 159 may be executed to navigate to and gather data from a single data source or from multiple data sources depending on pre-configuration. Conceivably, there is no limit to the number of URLs that may be included in one executed routine.

In still another embodiment, application 159 may be enhanced with additional enterprise rules to allow for even more intelligent parsing capability. For example, in addition to data reference libraries 163 and 165, which provide intelligence regarding text and style variations, an additional rules base may be provided for allowing system 161 to determine logical application of certain data-types that are parsed.

In some instances, data requested by a user may include time dependent data that may change or otherwise may not be valid after a certain period of time or date. One good example of this would be if a user requests all ongoing chat events from a plurality of separate chat sites wherein the events fall under a general topic or field of entertainment. Application 159 would navigate as previously described, and system 161 would parse the available chat titles at each site. An additional step would eliminate all chat-event titles or event fields that do not closely match the request thus narrowing the field. A subsequent step would eliminate all remaining events that are almost over. A next step may eliminate all remaining events that are marked for gender or age participation other than what is known about the requester. Ultimately, the field is narrowed to those events that match the request, have agreeable

participation requirements for the requester, and have sufficient time left in session for optimum participation. As a final step, the top 10 or so qualifying events may be selected for the requester based on number of participants etc.

The above method may be applied to any type of timed live event presented on the Internet. Live concerts, live video presentations, live radio shows, and many other types of WEB-events may be included. Rules governing such events may be compiled and made a part of user profile information and a rules base added to application 159 may be retrieved from that information and therefore will be individual to a requester. There are many possibilities.

Fig. 9 is a flow chart illustrating a semi-structured data retrieval process according to an embodiment of the present invention. In step 167 a user logs into his or her personal portal page. Typically, a user name and password is required during step 167 for authentication purposes, although in some cases the log-in may be automatic. At step 169 the user invokes application 139, which is presented as an interactive tool along with other functions in the portal page. Application 139 is used to pre-configure URLs for data search and return as was previously described. During this step, application 139 opens a browser window for navigation. It is important to note here that in a preferred embodiment, a user pre-configures each WEB page while on-line and navigating to each page. However, this is not required to practice the present invention. In one embodiment, a user may work off-line if he or she has all of the WEB pages cached in their presentable form.

At step 171, a user navigates to a desired URL for purposes of highlighting sections of the page for data retrieval. At step 173, the entered URL is displayed in the open window. Using a provided selection tool, a user then highlights a desired section or sections of a displayed WEB page at step 175. A user may highlight one or more sections of a same page. Moreover, one highlighted section may contain one or more sets of data a user wishes to retrieve. During step 175, a user also types in a data description concerning what type of data is to be returned from the highlighted portion of the page. If only one type of data exists in the highlighted section, then a description may not be necessary. It is noted herein that a user will likely highlight a

section larger than a portion just containing the desired data in case the data is moved slightly within a same WEB page between data searches.

At step 177 a user submits his or her selection and request entries to a server application responsible for further processing. Layer 143 of Fig. 7 represents the above mentioned server application. In this step, a user may configure many URL sites and data requests associated with the sites and submit all of the information collectively by one action.

At step 179, a DTD is generated for each highlighted portion of each URL. DTDs provide instruction to the parsing application as previously described in Fig. 7. At step 181, DTDs are organized and integrated with applicable site-logic including any log-in instructions required to access WEB sites. The result of this integration is an executable routine and instruction template for the navigation and data-gathering system taught in disclosure above. After a user submits all of the required data in step 177, the process is entirely automated.

At step 183, it is determined whether data results are requested immediately (on demand) or whether they are directed to be held for a user for later access. If a user requests immediate data return, then a browser control immediately executes according to generated instruction at step 191. A user may configure and request data from one URL at a time and receive data while still engaged in requesting data from a next page so as to operate in an active session of requesting and receiving data. However, typical implementation will be to pre-configure and request data from a series of URLs and schedule the data return for a later time.

At step 193, data is gathered and results are returned according to completed instruction. At step 195 a user receives the resulting data as it becomes available. It is noted herein that a user may practice an active session of immediate data return upon configuration and data request submissions. It is also noted herein that a user may complete a configuration and request submission process, and set-up a periodic time-table for the process to execute. The latter is the more likely and more useful scenario.

If it is directed that a user have his or her data results held for later access, then at step 185 a browser control is invoked for execution of the request. At step 185 data

is found and parsed according to instruction. At step 189, data results are aggregated and held for a requesting user. Activation of the data-gathering and return process may be delayed according to schedule with periodically scheduled return of results.

It will be apparent to one with skill in the art that the interactive process described herein may be altered to accommodate a number of different user preferences without departing from the spirit and scope of the present invention. For example, a user may pre-configure separate groups of URLs for data retrieval and cause separate processes to operate according to user selected time periods. For example, a user may wish to have all of his or her account information delivered on a certain day each week while he or she gets a news headline summary daily. There are many possibilities. The inventor intends that this process example represent just one of many possible orders for practicing the present invention.

Automatic Proxy WEB-Site/Service Registration

In another aspect of the present invention, a method and apparatus including software is provided for automatic proxy registration to a WEB-site or service on behalf of portal system subscribers.

Fig. 10 is a network overview of a communications network 197 enhanced with automatic site-registration according to an embodiment of the present invention. Communications network 197 comprises a data packet network (DPN) 199, a portal service domain 205, an ISP 201, and an exemplary user 203. Some of the elements that will be described in this specification are very similar to elements described in disclosure contained in the cross-referenced applications listed under the cross-reference section above. However, those similar elements re-introduced in this specification are modified or enhanced to practice the present invention. Therefore, similar elements newly introduced in this specification have new element numbers. Analogies are drawn to previous elements throughout this specification where appropriate.

DPN 199 is, in this example, the well-known Internet network and will hereinafter be referred to as Internet 199. Internet 199 is analogous to Internet 13 of

Fig. 1 in description and may well be another type of known DPN as described in Fig. 1 (Internet 13) above. The inventor chooses Internet 199 as a preferred example because of high public access characteristic. Internet 199 has an Internet backbone 207 illustrated therein. Backbone 207 represents all of the lines, equipment, and connection points that make up the global Internet network including connected sub-networks as are known in the art. Therefore, there is no geographic limit to the practice of the present invention. The inventor deems that illustration of just a portion of Internet 199 (illustrated within cloud) is sufficient for explanation and disclosure of the present invention.

Portal service domain 205 (enclosed in dotted rectangle) represents all of the equipment and network connections within Internet 199 that are hosted by an exemplary company providing portal services. For example, domain 205 comprises a WEB-server (WS) 217, a database server (DBS) 219, an instant server (IS) 221, and a navigation server (NS) 223. Each of servers 217-223 is network-connected to backbone 207. Moreover, all of the above-described servers are interconnected, in this example, by a separate data network 206. Network 206 is optional and only represents an efficient method for facilitating dedicated data and command communication between servers and is not required to practice the present invention. This example differs somewhat from the example of Fig. 1 wherein portal services are hosted by ISP 15. In this example, portal services are hosted within the domain of Internet 199 instead. The exact location of service-providing equipment is optional, as is the description of a hosting company.

WS 217 is configured as a main portal WEB-server and represents a first interface to subscriber/users exercising various portal services. Servers 219-223 represent separate secondary servers each providing dedicated function related to the present invention. For example, DBS 219 is adapted as a central repository for storing profile data, enterprise data, and summary data obtained from external sources within the domain of Internet 199 and returned thereto as described in the cross-referenced applications dealing with WEB summary gathering. IS 221 is adapted as an order generating server, which prepares job orders according to user requests, and NS 223 is

adapted as a proxy navigation server. More detail about servers 217-223 is provided below in this specification.

Illustrated outside of domain 205 are a WS 225 and a WS 227. WEB-servers 225 and 227 represent data or information servers as are known in the art for serving information and enabling services to any users registered thereto for purpose of obtaining access to data and services therefrom. For example, WS 225 may be a bank server, an investment server, an entertainment server, or any other class of servers known in the art. The same may be assumed regarding WS 227. A separate entity or entities other than the one hosting domain 205 in this example, typically host servers 225 and 227.

ISP 201 is configured to provide typical Internet connection services as are known in the art. User 203 is an exemplary user, which in this case, is a subscriber to the portal services available from domain 205. User 203 is provided access to Internet 199 through ISP 201. For example, user 203 operates an Internet appliance, illustrated herein as a personal computer (PC) 215. PC 215 is illustrated as having an Internet connection line 213 connected thereto and progressing to a modem bank 211 (shown as single modem icon) of an ISP connection server 209, which in turn is connected to backbone 207 within Internet 199.

In this way, user 203 has access to Internet 199. Internet connection line 213 may be a conventional telephone line, an ISDN line, a DSL line, or any other known connection technology including wireless connection. PC 215 may be any other type of known Internet appliance capable of connecting to and downloading information from Internet 199. In this example, it is assumed that user 203 is a subscriber to Internet portal services offered through domain 205. Therefore, it is assumed that an active session is established between user 203 operating PC 215 and WS 217 within domain 205. WS 217 is, in this example, a portal server adapted to serve a portal home page to user 203.

Also in this example, auto-registration capability is distributed over servers 217-223 within domain 205. However, this is not required in order to practice the present invention, but merely illustrated as such for purpose of discussion and separation of function. In other embodiments, other distribution arrangements may be

observed. Furthermore, all of the function of the present invention may be facilitated on one server.

WS 217 (portal server) has an instance of site configuration (SC) software provided therein. SC is a first part of an auto-registration suite. SC software provides an interface to user 203 from which he or she may pre-configure WEB-sites for auto registration. Part of that capability includes the provision of an interactive tool known to the inventor as "Reg Spy" (not shown). The basic function of SC software on server 217 is to allow user 203 to navigate to and select (using Reg Spy) the proper URLs and forms, which will then be used for obtaining registration to a site on behalf of user 203.

In one embodiment of the present invention, a plurality of WEB-sites may already be configured for auto-registration service and may be presented to user 203 at time of log-in to WS 217, perhaps, as part of his or her personal home page. This is made evident by illustration of an instance of portal host (PH) associated with WS 227. PH more accurately reflects that domain 205 already has all of the required site-logic and form data to allow a user to auto-register to a service provided by WS 227. PH, as illustrated here, could be present in the form of a hyperlink embedded in any given WEB-page held in WS 227 that is currently facilitating user registration. By clicking on a (PH) hyperlink during normal WEB navigation, user 203 could launch an auto-registration process beginning within domain 205. In some cases, however, user 203 may wish to add new sites for auto registration. WS 225 represents a server containing such a new WEB-site or sites not previously configured by domain 205.

DBS server 219, as previously described, holds profile data, enterprise data, and summary data. An instance of database renderer (DR) is provided to execute on DBS 219. DR software is adapted to allow writing to and parsing from DBS according to need. IS 221 has an instance of request generation (RG) software resident therein. RG software is provided and adapted to generate a machine-readable job-order from information submitted by user 203 and from prior information available in DBS 219. IS 221 formulates a job order, which includes instruction for navigation and log-in, if required. The output of RG software is a machine-readable

job-order that carries instruction for site-navigation and instruction for site registration purposes.

NS server 223 is adapted by means of an instance of auto-registration (AR) software. AR software is adapted to physically navigate (according to provided instruction) to a target site on behalf of user 203, and register user 203 to the target site or service. AR software performs a significant portion of auto registration in that it actually submits the data required for site-registration on behalf of user 203 and returns accepted user-data back to domain 205. Data "gathered" after an auto-registration function includes accepted log-in names and passwords to the site. NS software is similar to the "gathering agent" described in S/N 09/323/598 except that in addition to navigating to a WEB-site and retrieving data from that site, auto registration sequences may also be performed.

Total auto-registration capability includes a means for site-configuration (SC), a means for converting collected data and pre-known information into a job order (RG), a means for proxy navigation and data submission to a WEB-site (AR), and a means for returning successful registration data to a data repository for subsequent use (AR, DR).

As previously described, WS 225 and WS 227 represent WEB-servers containing URLs invoking a form or forms, which user 203 may fill out (populate) and submit for purpose of obtaining services therefrom. In addition to registration forms, other forms may be processed by method and apparatus of the present invention. For example, purchasing forms, survey forms, classified add forms, auction forms, and others make up a list of possible form types to which the methods and apparatus of the present invention may be adapted.

Auto registration may follow two separate orders. One is that a user selects a pre-configured site for registration. The other is that a user pre-configures a found site for registration. Assume now that user 203 wishes to register to receive a service available from WS 225. In this case, WS 225 is not pre-configured or necessarily known to domain 205. In order to auto-register to a site or service hosted by server 225 user 203 must first pre-configure the site.

In general practice of the present invention, user 203 would first log-in to WS 217 and establish an on-line session between PC 215 and WS 217. User 203 would then invoke SC software, and proceed to navigate to WS 225 and to the page offering registration for services. Upon arriving at the site, user 203 invoking SC software, highlights the required form or forms that must be filled for registering to a service or services facilitated by the target WEB-site, and page keywords identifying the form carrying page to the service. This "site data" including form data is transferred to WS 217.

Once WS 217 has the required site data including the required forms, it checks DBS 219 to see if all of the required "submission" information is already available in a user profile to submit for this particular registration action. It is noted herein that user 203 may have several stored profiles that he or she may use to register to different types or classes of services. In this case a user may designate a particular profile for registering to a particular service as part of the configuration process. Data stored in a profile may include credit card information, address information, e-mail information, telephone information, age and gender information, monikers (aliases), and so on. If a particular item of data is required, but not found in a designated profile, user 203 may be prompted by WS 217 to submit the required snippet of data.

Of course, auto registration is intended to eliminate or at least reduce to a minimum the repetitive data-entry normally required to fill out numerous forms at different WEB-sites. Moreover, repetitive submission of user-chosen passwords, log-in codes, and user-names for site acceptance or denial can be eliminated. In a preferred embodiment, user-names, log-in codes, passwords, and the like are generated and submitted by the auto-registration process. This can be accomplished under partial user direction, or it can be accomplished completely without user involvement. In the first case, user 203 may have a list of unused passwords, user-names, and/or log-in codes that were pre-chosen and stored along with profile information. In the second case, such submissions may be automatically generated and submitted according to form requirements and site rules wherein the user has no knowledge of the data even after successful registration. The second embodiment

follows a scenario wherein passwords and the like are optionally managed by the password-all portal service of application S/N 09/208,740.

Referring again to Fig. 10, all of the site data is passed to IS 221 over data network 206 (optional). In another embodiment, the data is passed over shared bandwidth on backbone 207. IS 221 is responsible for request generation (RG). Applying all of the required site data, navigational data, and profile data, IS 221 creates a job-order (not shown) containing complete instruction for navigation, form filling, and data return. Applicable user data is encrypted when transferred over secure socket links (SSL) as is known in the art.

In a preferred embodiment, job orders are written in eXstensible Mark-up Language (XML), however, this in not required in order to practice the present invention. Other suitable mark-up languages may also be employed. A completed job order is then passed to NS 223 over network 206 or backbone 207. NS 223 using AR software executes the job order according to instruction.

NS 223, executing a job order on behalf of user 203, navigates to WS 225, in this example, and submits data either directly from DBS 219 where it is stored, or from within the order itself. The first case assumes that codes are used in the XML order that equate to actual values in DBS 219. In this case, the data is not released from DBS 219 until it is requested by NS 223 during the form-filling process. The second case assumes that the XML order includes the actual values. In this case, the portion of the XML order containing user information is encrypted before navigation.

After successfully registering user 203 to obtain services offered by WS 225, accepted log-in and any password data is returned to DBS 219 and written thereto as part of a new user-site block pertaining to that particular WEB-site or service. Such a site may now be included in future data-gathering and summary processes wherein auto-log-in is possible as described in cross-referenced application S/N 09/323,598.

In a case wherein a site is already known to domain 205 and is already pre-configured for navigation and form filling, as is the case with WS 227, the process follows a different order. For example, user 203 would log-in to WS 217 (portal server) and receive his or her personal portal page as described above. User 203 may either select a service hosted by WS 227 from a presented list of services, or he or she

may navigate to the actual WEB-site where an icon (PH) is visible, such as on a registration page of the site.

In some cases, a user may simply be browsing WEB pages and stumble upon a registration icon. By clicking on the icon, or a similar representation thereof, while visiting the site or while browsing a list of pre-configured sites within his or her portal page, the auto-registration process is automatically launched from within domain 205. The process is essentially identical, except that much of the required data such as site navigation data and form data is already known to domain 205. Therefore, several steps may be eliminated from the process as a whole. Moreover, a generic XML order may already exist for services hosted by WS 227 such that only the profile data generic to a specific requesting user need be added to the order.

It will be apparent to one with skill in the art that the method and apparatus of the present invention may be practiced in a variety of ways without departing from the spirit and scope of the present invention. For example, sites offering services may be pre-configured for auto registration services by domain 205 such as is the case with WS 227. In these scenarios, such sites and services may be researched and rated ahead of time as to quality of service including dependability, success rate, customer service capability, and so on. By performing much of the pre-registration research generally required of a customer wishing to register for any services, a customer may be put more at ease and may be more likely to do more business on the Internet. The fact that a third party, which is already trusted by users, performs auto registration extends that credibility to the particular site offering the services.

Still, a user may desire to find and register sites (WEB findings) that are not known to domain 205. As part of the service offered by domain 205, users understand and accept any risks posed by added sites as part of normal business. Registration by proxy on behalf of requesting users is possible for virtually any offered service that might be obtained through the Internet by interactive form.

Fig. 11 is a block diagram illustrating software architecture of an auto-registration suite 229 according to an embodiment of the present invention. Suite 229 represents the server-distributed software components of Fig 10 above. As previously described suite 229 may be divided to execute on a plurality of machines, or it may be

provided on one powerful server. In this example, all described components are grouped by layer into one application or suite 229 for exemplary purpose only.

Suite 229 has a software layer 231, which is provided therein and adapted as a site-configuration layer (SC). Layer 231 is analogous to SC software illustrated on WS 217 of Fig. 10. Layer 231 has a portal interface module 239 provided therein and adapted to function as a user-interface for general portal services as well as an interface for services related to auto-registration services. For example, a user may first invoke module 239 in order to cause display of his or her portal options.

A registration spy (Reg-Spy) module 241 is provided within layer 231 and is user-accessible through portal interface 239. Reg-Spy 241 is a tool that enables a user to navigate to any service-offering site for the purpose of highlighting site forms and other associated data that is required for successful navigation and registration to a site or service. Reg-Spy 241 may be used to highlight a main registration form and description, as well as any secondary forms and descriptions required to receive secondary services, added services, optional services, or customized versions of main services. By default, the auto-registration service of the present invention will only register users to main services linked from pre-configured sites. However, a plurality of services, or options depending from a main service may be configured manually for auto-registration if desired.

A database interface module is provided within layer 231 and adapted for enabling bi-directional database communication with DBS 219 of Fig. 10. A profile generation module 245 is provided within layer 231 and adapted to enable a user to create and add a new user profile to a list of existing user profiles. Using module 245 in conjunction with module 243, a user may also edit existing profiles, or delete existing profiles. In this example, modules 241-245 are secondary modules available through portal interface module 239, however that is not specifically required in order to practice the present invention. For example, a user may log-in solely for the purpose of editing a profile, in which case, module 245 may be requested and presented as a first option after log-in. Similarly, module 241 may be requested as a first option after portal log-in. In normal practice of the present invention, various options would appear on a user's personal portal page as previously described.

BNSDOCID: <WO _____ 0180067A1 I. >

Suite 229 has a software layer 233 provided therein and adapted for request generation (RG) and activation, as so labeled. Layer 233 provides function, which is analogous to the function of IS 221 of Fig. 10. Layer 233 has an auto-registration module 247 provided therein and adapted as a software mechanism for initiating a request generation process for registration to a particular WEB-site. Module 247 may also be provided within layer 231 without departing from the spirit and scope of the present invention. For example, a number of pre-configured services may be listed and accessible from interface 239 wherein one may be selected and then activated to target level by invoking a list-associated module 247 thereby launching the request generation process.

Layer 233 has a database interface module 249 provided therein and adapted for bi-directional database communication as described above with module 243. Within layer 233, module 249 allows access to DBS 219 of Fig. 10 for the purpose of verifying that all of the required data is available at the time of request generation. It is noted herein that profile data held in a user profile is not necessarily configured for insertion to a particular form according to site rules. Therefore, a base converter module 251 is provided within layer 233 and adapted to "convert" profile data to a form acceptable to required form fields using the applicable format required by the target site at the time of registration.

Base converter module 251 may also comprise several sub-modules (not illustrated), which are adapted to handle specific types of data. For example, age requirements may be handled by a "birthday converter", financial payment data may be handled by a "payment converter" and so on. Module 251 is responsible for formatting available data according to site and enterprise rules. As such, input and output dependency tables are employed as a method of understanding various requirements for data formatting.

In order for the service in an embodiment of the present invention to register a user to a site, it must first obtain all of the service-dependent profile fields required for successful registration at a particular site/service. To accomplish this, all of the required site-specific forms must be obtained along with all of the site-specific fields

contained therein, as well as the specific data-format requirements for each field. This is accomplished during site configuration using module 241.

An input dependency table is created listing all of the user fields and format requirements needed by a particular site for registration. A second list is obtained that contains all of the user fields of the input dependency table that a user already has stored in his or her profile data table. Such a table may hold items that are used in different profiles such that the list is a compilation of all of the stored data items repeated throughout several separate profiles without repeating one of the data items.

By comparison then, a list of fields missing data is developed if there are any. If there are no required fields missing data, a request may proceed from WS 225 to IS 221. Converter module 251 insures the data is properly formatted for site-specific rules. If data is missing from a user profile, then the user is prompted to supply the missing data.

The job of converter module 251 is to map user data into a format accepted by a particular form. For example, a user profile format for a full name may read last name, middle name, and first name. This format is widely recognized in many formal applications however, some sites will require a first name followed by a last name with no space for a middle name or initial. Using Reg-Spy tool 241, a user has highlighted the target form-field, and the rules that apply. Therefore, at the time of request generation in layer 233, converter module 251 has the required format requirements for that specific section of data. When accepting user information for profile generation, the service may ask for more than is typically required on most forms. In this way, the chance that a user will be required to add something after a request generation has initiated is minimized.

Layer 233 has a job-generation module 253 provided therein and adapted to write a functional XML instruction order containing all of the required data for navigating to and registering a user to a particular service. Base converter 251 supplies all of the re-formatted (if required) user-data to module 253 for inclusion into the order. A job order will contain a site-navigation template, an auto-login block (if required), and user data or a reference to the appropriate user data to submit to site forms required for registration purposes. A test module (not shown) may be built into

an XML order generated by module 253 such that form filing may be rehearsed for accuracy before actually submitting the data to a site. Such a test module may instead be provided as part of Reg Spy 241. Once an XML order is deemed correct and is ready to execute, it is passed on to a navigation sub-system through a navigation interface module 255.

After data conversion and all other requirements are satisfied, an XML job order written by module 253 will resemble the following exemplary order.

```
<register url="http://lc2.law5.hotmail.passport.com/cgi-
bin/tos.cgi?us=hotmail&amp;_lang=&amp;beta=">
        <form>
                <action type="fill_and_submit" actionArg="Submit.x">
                        <keywords>Profile Information</keywords>
                </action>
                <field type="OPTIONS" name="month">
                        <value>03</value>
                </field>
                <field type="PASSWORD_CONF" name="passwd1">
                        <value>5cacb3cb76ab4e7996aff6fa630412e4</value>
                </field>
                <field type="OPTIONS" name="day">
                        <value>31</value>
                </field>
                <field type="TEXT" name="xage">
                        <value>21</value>
                </field>
                <field type="OPTIONS" name="xoccupation">
                        <value>un</value>
                </field>
                <field type="LOGIN" sendback="1" name="reglogin">
                        <value>ji</value>
                        <value>yodleeuser</value>
                        <value>ji34</value>
                        <value>yodleeuser43</value>
                </field>
                <field type="PASSWORD" sendback="1" name="passwd">
                        <value>5cacb3cb76ab4e7996aff6fa630412e4</value>
                </field>
                <field type="TEXT" name="xfname">
                        <value>Ji</value>
                </field>
                <field type="TEXT" name="xlname">
```

```
                <value>Lee</value>
            </field>
            <field type="TEXT" name="xzip">
                <value>94086</value>
            </field>
            <field type="RADIO" name="xgender">
                <value>m</value>
            </field>
            </form>
            <form>
        <action type="reset_auto_reg"actionArg="">
            <keywords>someone has already selected</keywords>
        </action>
            </form>
            <form>
        <action type="fill_and_submit" actionArg="I Accept">
            <keywords>MSN HOTMAIL TERMS OF SERVICE</keywords>
        </action>
            </form>
            <form>
        <action type="reset_auto_reg" actionArg="">
            <keywords>Hotmail Registration</keywords>
            <keywords>Sign-In Name contains reserved or ineligible
        word</keywords>
        </action>
            </form>
            <form>
        <action type="success" actionArg="">
            <keywords>Sign Up Successul</keywords>
        </action>
            </form>
            <form>
        <action type="reset_auto_reg" actionArg="">
            <keywords>Sign-In Name already in use</keywords>
        </action>
            </form>
</register>
```

The exemplary order shown above has been slightly altered in indentation for some lines to compensate for page margin. One with skill in the art, however, will recognize the proper XML format. It is also noted herein that values listed in the order do not contain actual data, but coded equivalents to the data as registered in DBS 219 of Fig. 10. In this embodiment profile data is only pulled when required for submission as previously stated. In this way, security is enhanced and the actual order

may be condensed. In one embodiment, metadata may be inserted into value fields contained in a job order.

Suite 229 has a software layer 235 provided therein and adapted for auto-registration (AR) and data refresh. Layer 235 is responsible for navigating to a target WEB site and submitting registration information to the site on behalf of a requesting user. Upon successful registering, layer 235 is also responsible for the return of confirmation and log-in information to a user and database

Layer 235 has a navigation engine 257 provided therein and adapted to navigate to a target WEB-site based on navigation instruction contained in a job order generated in layer 235. Navigation engine 257 may navigate to any site or series of sites specified in the associated job order. Engine 257 also performs auto-log-in procedures when required. A parsing module 259 is provided within layer 235 and adapted to parse information from selected portions of WEB pages during navigation. Therefore, module 259 works in conjunction with and may be a part of engine 257.

It is taught in the cross-referenced application entitled ***"Method and Apparatus for Retrieving Information From Semi-Structured, WEB-Based Data Sources"***, that a parsing function parses all data from user-highlighted sections or blocks of data contained in target WEB-pages from which meta-summaries will be compiled. The parsed data is compared to user-input data regarding the nature of the data request. When parsed words match keywords contained in the request, the associated data is returned.

Parsing module 259 operates in much the same way in that it looks for page keywords, and form requirements highlighted by a user during pre-configuration. Once a page and section (form) has been identified, registration action continues as specified by the job order generated in layer 233. A database interface 261 is provided within layer 235 and adapted to allow bi-directional communication to DBS 219 of Fig. 10. In this way, the appropriate data may be ordered for submission.

A data writer module 263 is provided within layer 235 and adapted to transfer the appropriate data into the appropriate form fields in the format required by the particular form. At this point, a completed form is submitted to a service-hosting site. In some cases, chosen user-names, log-in codes, etc. will not be accepted by a site. It

may be that another user has already chosen a particular name. In this case, a retry attempt is initiated using a different value either generated by the service, or pre-chosen by a user as previously described.

A data return module 265 is provided to return successful registration information including user notification thereof and record of successful and accepted values submitted for future log-in purposes. In some cases, accepted values may be immediately used by the service to log-in on behalf of a user and to obtain data from the site for a user if directed to do so by XML order.

Suite 229 has a database refresh layer (DBS) 237 provided therein and adapted to accept return information from layer 235 for the purpose of recording the information for future use. A database filer module 267 is provided within layer 267 and adapted to create a new member item for user 203 (Fig. 10) and to write the returned data thereto. In this way, the new site is included in a list of user-registered sites for data gathering and summary presentation taught in the cross-referenced applications.

A data validation module 269 rechecks all data written into DBS 219 (Fig. 10) and supplies any new data to an input dependency table described previously. In some cases, some user-supplied data will not be valid after a successful registration. For example, if a phone number or other temporary information is supplied in conjunction with registration to a service, it may be deleted after it is no longer needed for registering. One example would be that of a temporary site registration. In this case all of the new data supplied for use with a temporary site registration may be purged from DBS 219 after the time period allotted for the service has expired. Such a service may be a limited free period of technical support for a purchased product or the like.

A user notification module 267 is provided within layer 237 and adapted to notify a user upon successful registration and in some cases failed registration attempts. It may be that a site disqualifies a particular user from registration based on information such as credit rating or some other criteria. In this event, a notice would be sent to the user. A user presentation module 273 is provided and adapted to present any summary or refresh data to a user if it was requested before registration.

Such data may be returned along with registration confirmation and log-in data. Module 273 is also used during normal summary gathering operations after registration and inclusion of the target site into a user's list of registered sites.

It will be apparent to one with skill in the art that the software components contained in layers 231-237 of suite 229 may vary in number and description without departing from the spirit and scope of the present invention. For example, instead of multiple database interfaces, one database interface may be provided for a suite 229 that is running on one server. Moreover, layer 235 may comprise more than one of each described function for situations wherein an XML order is split or distributed among several navigation, auto registration, and data gathering applications, combined to define "gatherers" as termed by the inventor. An example would be that of a single XML order containing a plurality of URL sites and registration requirements for several of the included sites. Separate gatherers may be used to execute several actions contained in the order. The inventor intends that the example presented herein represent just one possible software-architecture that may be used to practice the methods of the present invention.

As previously described, the layered structure illustrated herein with respect to layers 231, 233, 235, and 237 and components represented therein may be distributed in parts over various servers, or contained as one application running on one machine. There are many possibilities.

Fig. 12 is a process flow diagram illustrating various process steps for auto-registration to a site and receiving data from the site according to an embodiment of the present invention. At step 275, a user logs-in to a portal server to obtain portal services. In this step a portal page is downloaded to the user. It is assumed in this example that all site pre-configuration is complete and all data is available for navigation and registration purposes. It may be that the user is adding a new site or sites that are not pre-known to domain 205. It may be that site or sites selected by a user are pre-known to domain 205. In some instances, it may be a combination of the above. In case of any of the above conditions, a user invokes an auto-registration icon at step 277. Profile selection, as described above, is part of the pre-configuration process and therefore, not represented in this process flow.

In one embodiment, a separate registration icon is associated to pre-configured WEB-sites (one per) appearing in a list of service-offering sites on a user's portal page. In another embodiment, one registration icon is invoked after a user has selected more than one service-offering site for registration. In still another embodiment, a user may come across such an icon while normally surfing the Internet, at which time he or she may invoke it to launch an automated registration process.

At step 279 the portal server of step 275, also termed a WEB-server accesses a database to check if all of the information about the user is available. It is assumed in this example, that all data is available. If some data were not available, a user would be prompted to submit it. At step 281 all user profile data and acquired site data is verified as authentic. In this step, codes may be assigned to stored data for latter retrieval based on such codes. At step 283, user data is converted to site data with respect to any format required by site-specific form-fields. At step 285, an XML order is generated based on all required data to complete a proxy navigation and auto-registration process.

At step 287, the job order generated in step 285 is deemed complete. This step may include data verification and a self-test sequence to insure that all of the user data can be correctly mapped into the appropriate form fields. If an error occurs during a self-test sequence, then a user is notified of the error and an attempt to correct the error is initiated.

Once a job order is deemed complete, it is passed to a navigation sub-system in step 289. The navigation system receives and executes the job order according to instruction. At step 291, a target WEB-site is accessed according to navigation instruction and page keywords. Page keywords are used to verify the exact page containing the required form for form filling.

In one embodiment, part of the job order described above contains a DTD template interwoven into site logic as described in Fig. 8 above. Using this method allows a form to be located based on HTML data-structure matching. The target form is highlighted during pre-configuration and a DTD is produced for that section and for the URL wherein the section resides.

At step 293, a target form is populated with the appropriate user data and submitted. This is accomplished according to the action portion of the job order. At step 295, the registration attempt of step 293 either succeeds or fails. If in step 295, a form is accepted, then in step 297, the appropriate data and confirmation is obtained and returned.

If in step 295 the registration attempt fails, an attempt to determine the reason for failure is initiated. In this step, notification may be sent to a user explaining the given reason for failure. It may be that the user just does not qualify for registration to the target site or service. It may be that an error has occurred with respect to data submitted such as a credit card number is invalid or expired. In most cases however, failure on a first attempt will likely be the result of another user already possessing a submitted password, user-name or log-in code.

If a failure notice is displayed by a target service to the navigation system, it may be parsed for content with notification sent back to a user. In this case, a user may be required to update his or her selected profile with correct data, or select an alternate profile and re-initiate the process.

If the failure notice involves submitted passwords or the like already being taken, a re-try using new data is attempted by default. This is represented by step 299. This step may be repeated many times as new data is obtained to replace data rejected by a site. As previously described, user-names, passwords, and log-in codes may be supplied in the form of a list selected and approved by a user, or may be generated at the time of form population. In step 301, registration succeeds and in step 303, the accepted password and log-in data along with confirmation data is returned to a user.

It will be apparent to one with skill in the art that the process described herein may contain additional steps and sub-steps without departing from the spirit and scope of the present invention. For example, step 291 may contain a series of sub-steps such as parsing, structure matching, and so on. As well, processes related to registration error, user notification thereof, data correction and re-submission may be tailored somewhat based on the type of error communicated.

The method and apparatus may be practiced with a wide variety of interactive forms other than site registration forms without departing from the spirit and scope of the present invention.

The method and apparatus also may be practiced in a language and platform independent manner, and be implemented over a variety of scaleable server architectures.

The method and apparatus of the present invention may be practiced via private individuals on the Internet, businesses operating on a WAN connected to the Internet, businesses operating via private WAN, and so on. There are many customizable situations. The present invention as taught herein and above should be afforded the broadest of scope. The spirit and scope of the present invention is limited only by the claims that follow.

What is claimed is:

1. A software application for populating and submitting interactive forms by proxy, comprising;

a function for finding and capturing data about a site associated with the form and about the form associated with the site;

a function for writing a job order containing data specific to the site, the associated form, and a requesting user;

a function for navigating to the site and submitting data to a host sponsoring the site using the form associated with the site; and

a function for returning and recording a portion of the form-submitted data accepted by the host for subsequent use in gaining access to the site.

2. The software application of claim 1 wherein the site is an interactive information-page located on a data-packet-network.

3. The software application of claim 2 wherein the site is an electronic WEB page and the data-packet-network is the Internet network.

4. The software application of claim 2 wherein the form is an interactive registration form contained within the site.

5. The software application of claim 4 wherein the host is a company providing a service available through the site and accessible through the form contained in the site.

6. The software application of claim 3 wherein all of the software functions are resident in a single server-node on the Internet and are executed from the single server-node.

7. The software application of claim 3 wherein certain software functions

are distributed over a plurality of server nodes on the Internet and are executed from those server nodes.

8. The software application of claim 3 wherein the job order is written in the form of eXtensible mark-up language.

9. A hardware-software system for configuring and executing a job order for populating and submitting an interactive form to a host and returning data from the host comprising;

a portal server running configuration software for interfacing with a user and for accepting data input in the form of an order request and configuration data;

a generation sever running order-writing software for generating the job order according to a job request initiated by the portal server;

a navigation server running form population software for executing the job order received from the generation server according to instruction and data contained therein; and

a data repository accessible to the servers for recording, storing and serving data, such that a user connected to the portal server and utilizing the configuration software may obtain data about a site and about an associated interactive form, the data ultimately made available along with user data to the generation server which writes the job order for execution by the navigation server, and wherein accepted form-submitted data is returned by the navigation server to the data repository and recorded for subsequent access to the site.

10. The hardware-software system of claim 9 wherein the host is a company maintaining an interactive information page located on a data-packet-network.

11. The hardware-software system of claim 10 wherein the interactive information page is an electronic WEB page available on the Internet.

12. The hardware-software system of claim 11 wherein the form is an

interactive registration form provided in the electronic WEB page.

13. The hardware-software system of claim 12 wherein the job order is written in eXtensible mark-up language.

14. The hardware-software system of claim 10 wherein a separate and dedicated data network inter-connects the server-nodes for communication.

15. A method for populating and submitting an interactive form by proxy comprising the steps of;

(a) obtaining data specific to a form, a site, and a requesting user;

(b) writing a job order for navigating to the site for the purpose of populating and submitting the form to a host associated with the site; and

(c) executing the job order.

16. The method of claim 14 wherein in step (a), the site is an electronic information page hosted on the Internet and the form is an electronic form located within the information page.

17. The method of claim 14 wherein in step (b), the job order is written in eXtensible mark-up language.

18. The method of claim 14 wherein step (c) includes a sub-execution for obtaining data from the site after successful form submission to the host sponsoring the site.

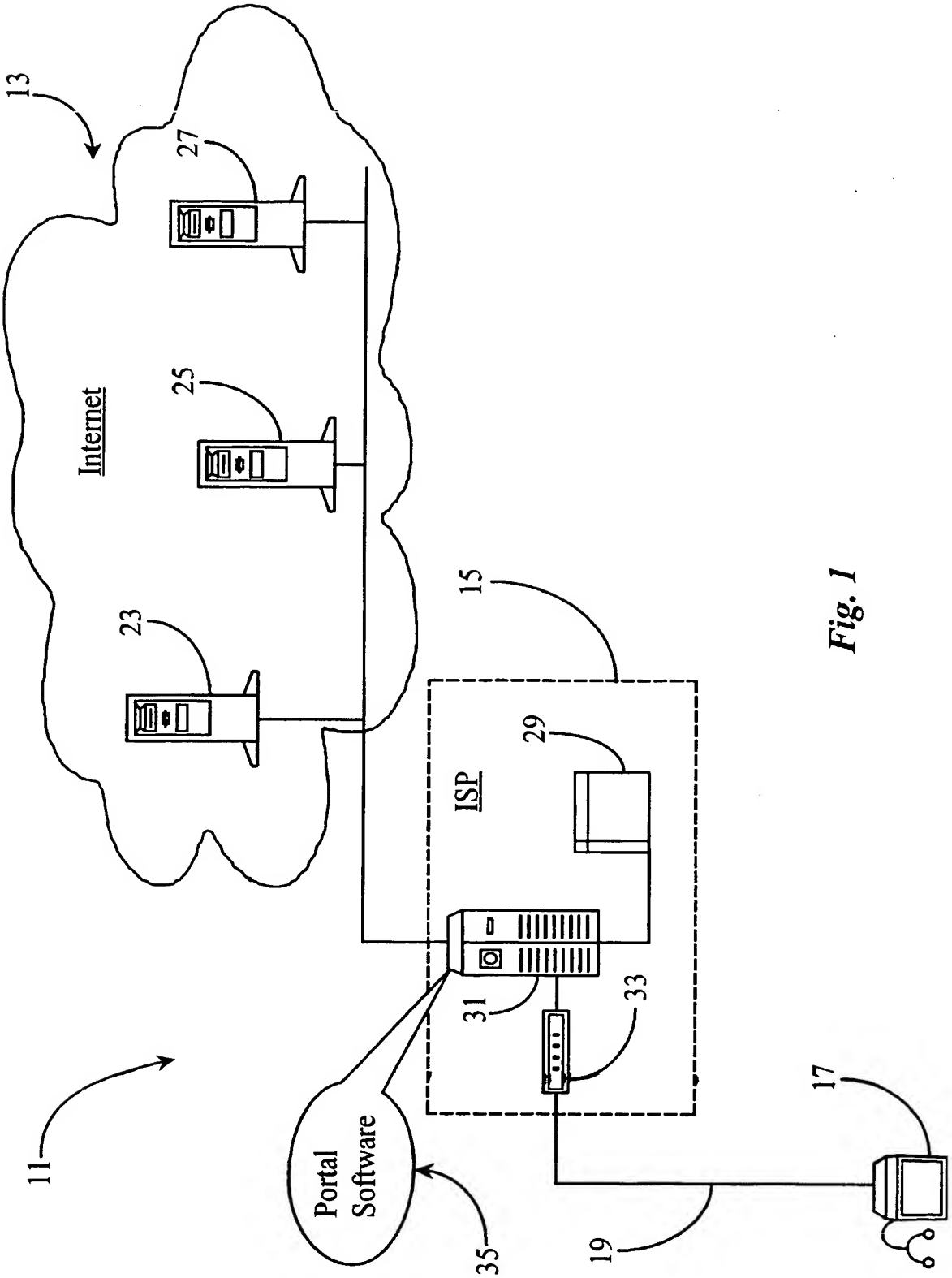


Fig. 1

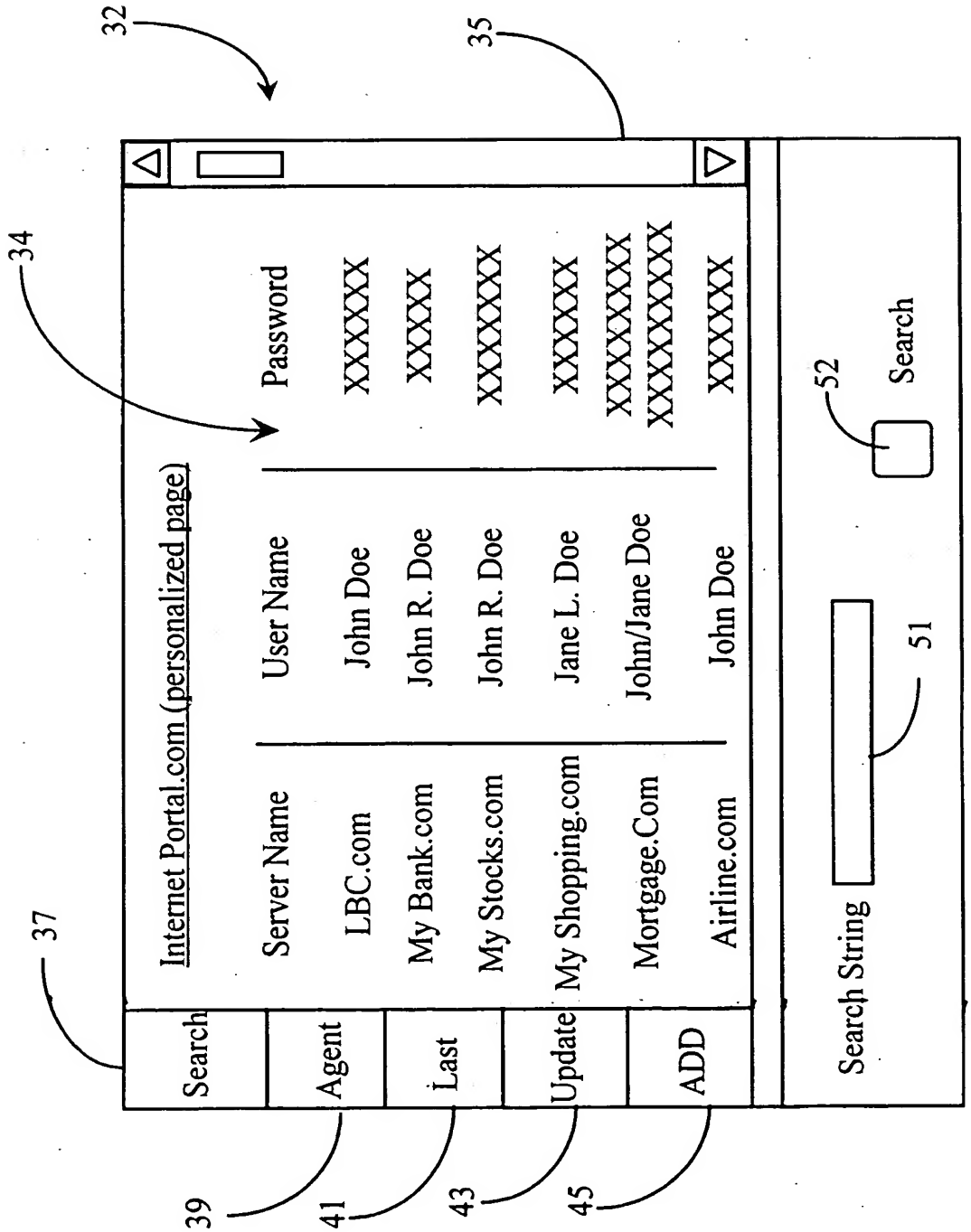


Fig. 2

BNSDOCID: <WO_____0180067A1_I_>

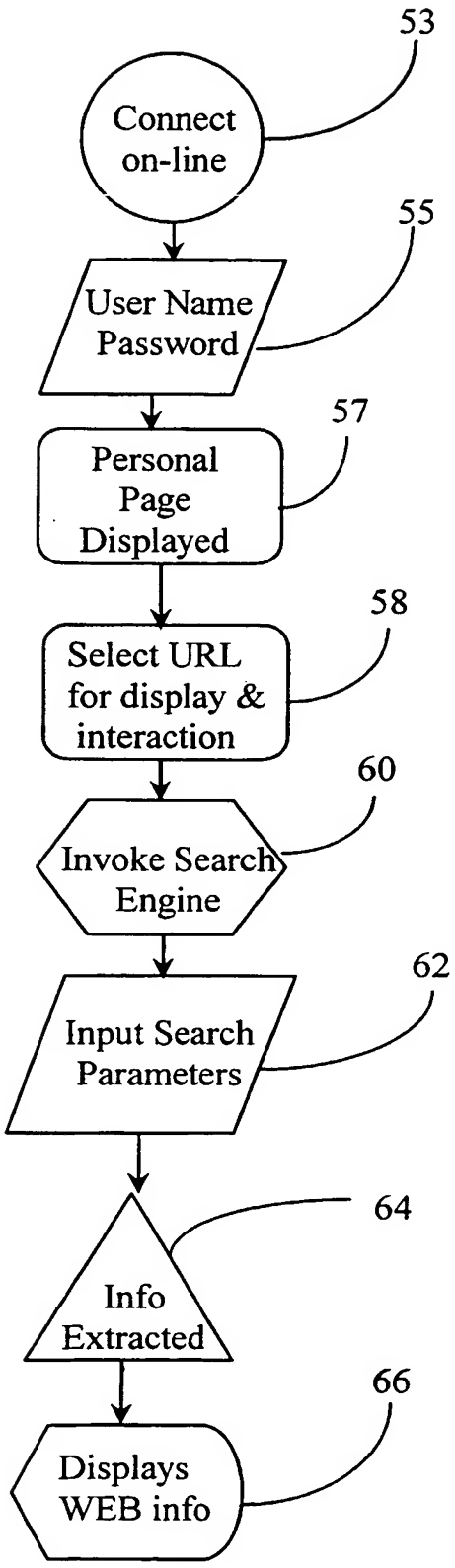

53
Connect on-line
55
User Name Password
57
Personal Page Displayed
58
Select URL for display & interaction
60
Invoke Search Engine
62
Input Search Parameters
64
Info Extracted
66
Displays WEB info

Fig. 3

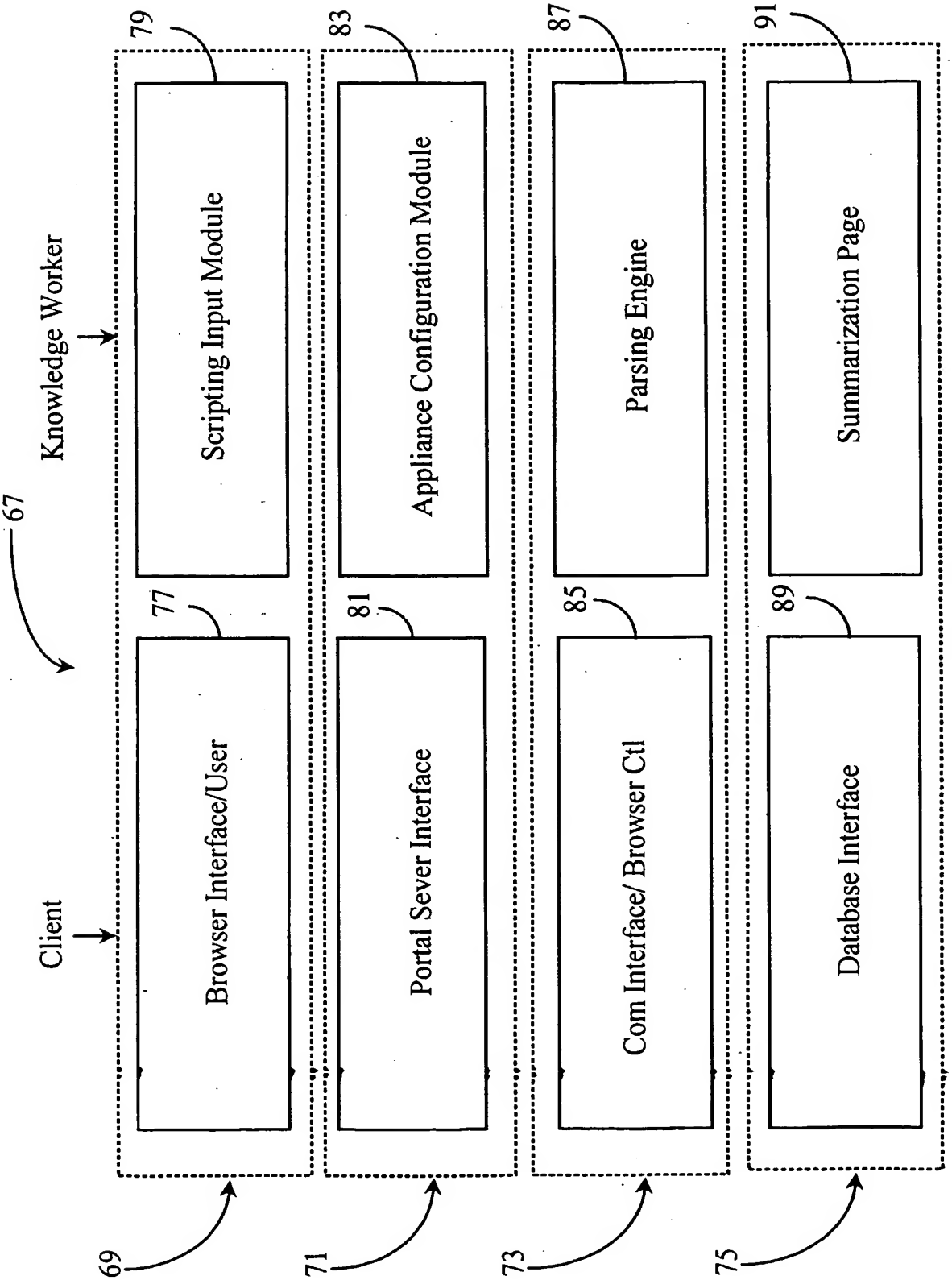


Fig. 4

BNSDOCID: <WO_____0180067A1_I_>

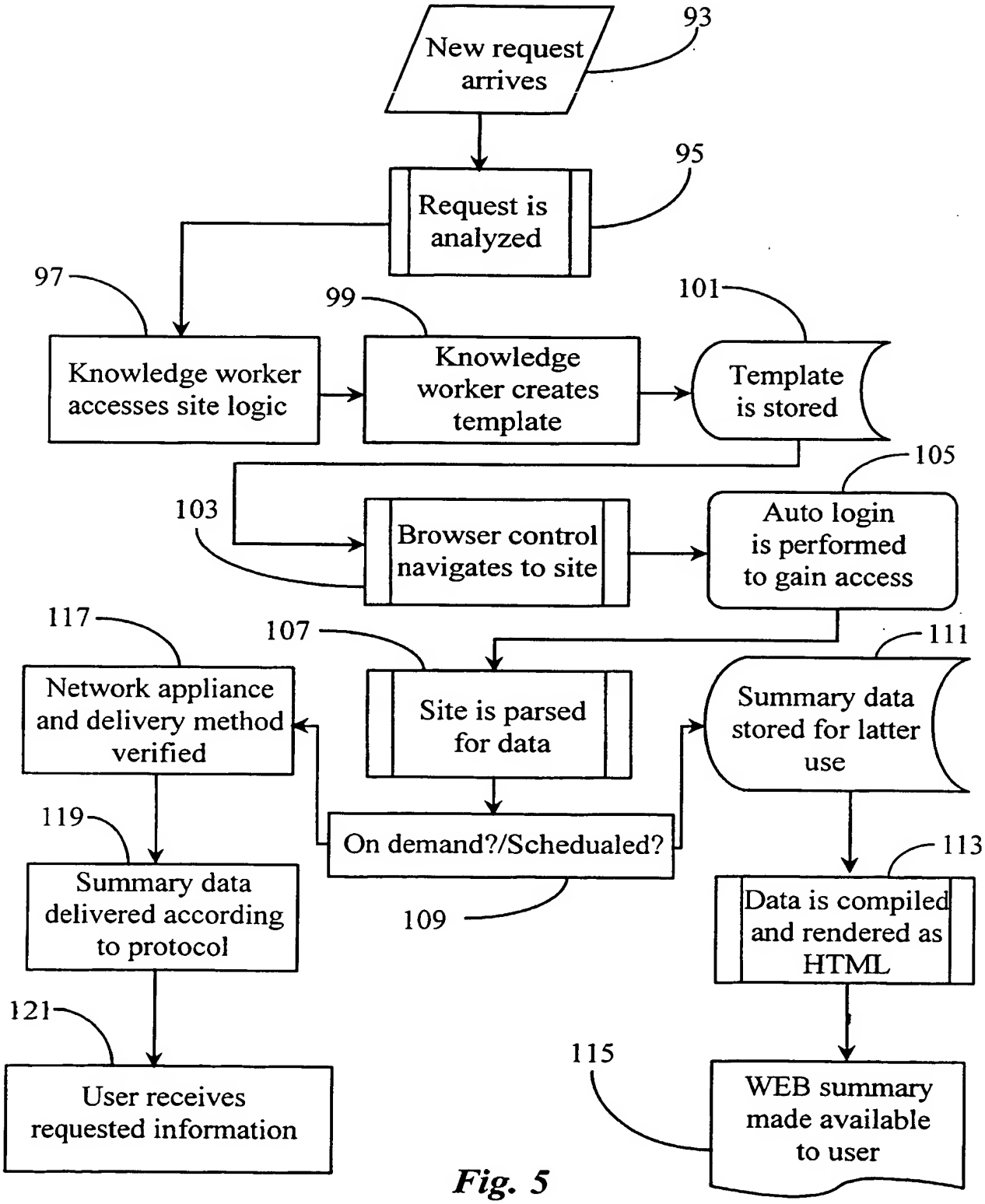

New request arrives
93
Request is analyzed
95
97
Knowledge worker accesses site logic
99
Knowledge worker creates template
101
Template is stored
105
103
Browser control navigates to site
Auto login is performed to gain access
117
107
111
Network appliance and delivery method verified
Site is parsed for data
Summary data stored for latter use
119
On demand?/Schedualed?
113
Summary data delivered according to protocol
109
Data is compiled and rendered as HTML
121
115
User receives requested information
WEB summary made available to user

Fig. 5

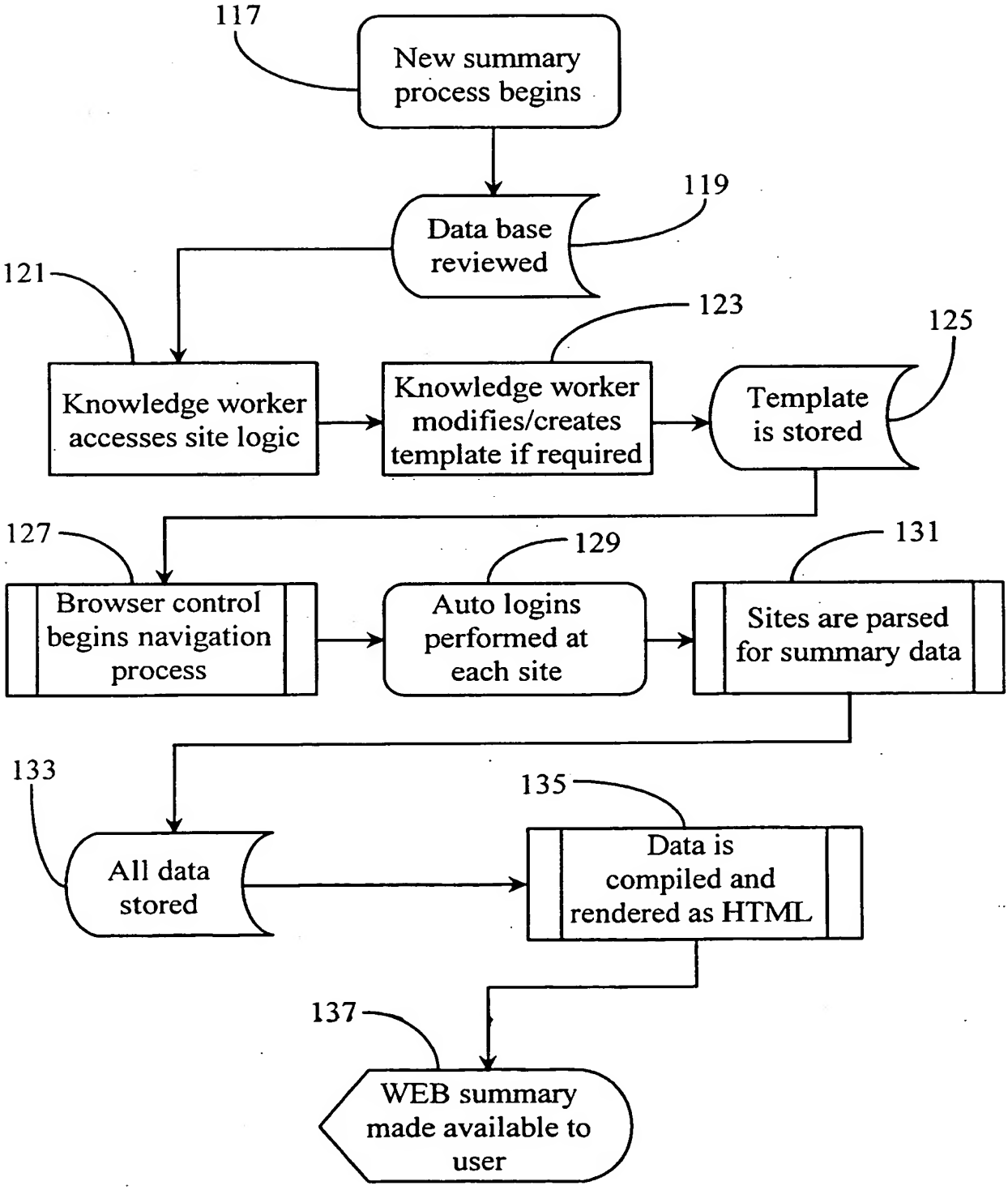

117
New summary process begins
119
Data base reviewed
121
123
125
Knowledge worker accesses site logic
Knowledge worker modifies/creates template if required
Template is stored
127
129
131
Browser control begins navigation process
Auto logins performed at each site
Sites are parsed for summary data
133
135
All data stored
Data is compiled and rendered as HTML
137
WEB summary made available to user

Fig. 6

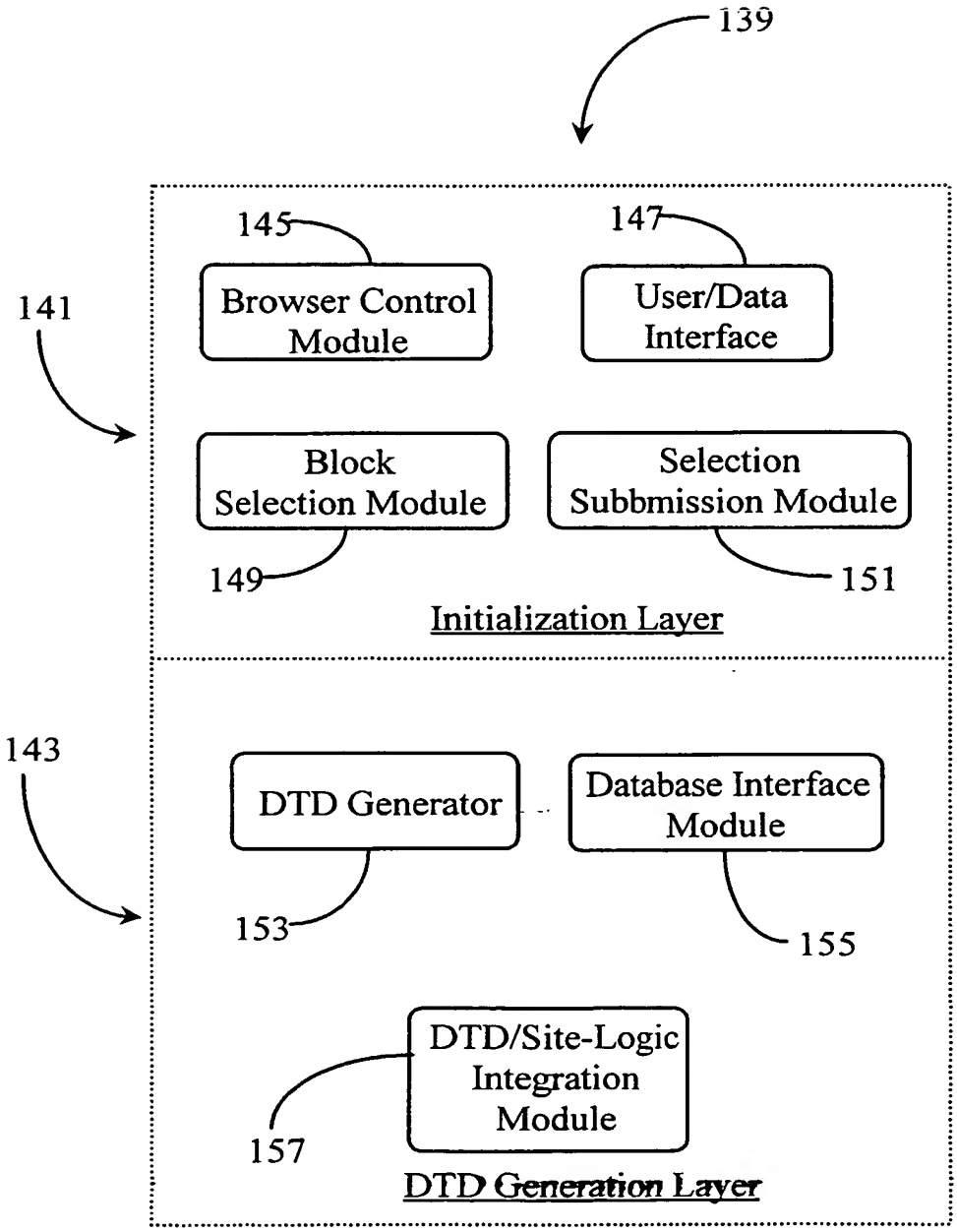


Fig. 7

BNSDOCID: <WO_____0180067A1_I_>

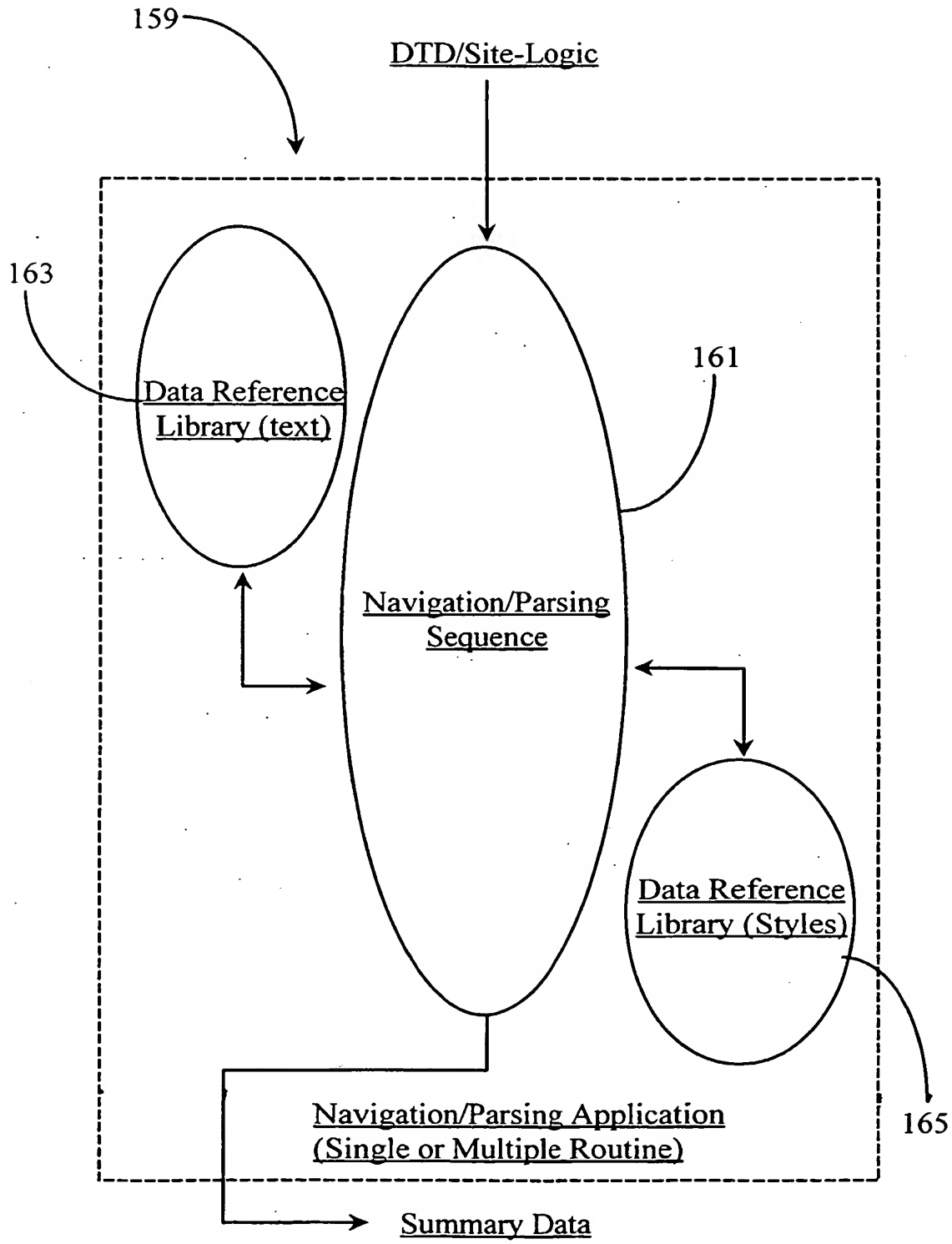

159
DTD/Site-Logic
163
Data Reference Library (text)
161
Navigation/Parsing Sequence
Data Reference Library (Styles)
165
Navigation/Parsing Application (Single or Multiple Routine)
Summary Data

Fig. 8

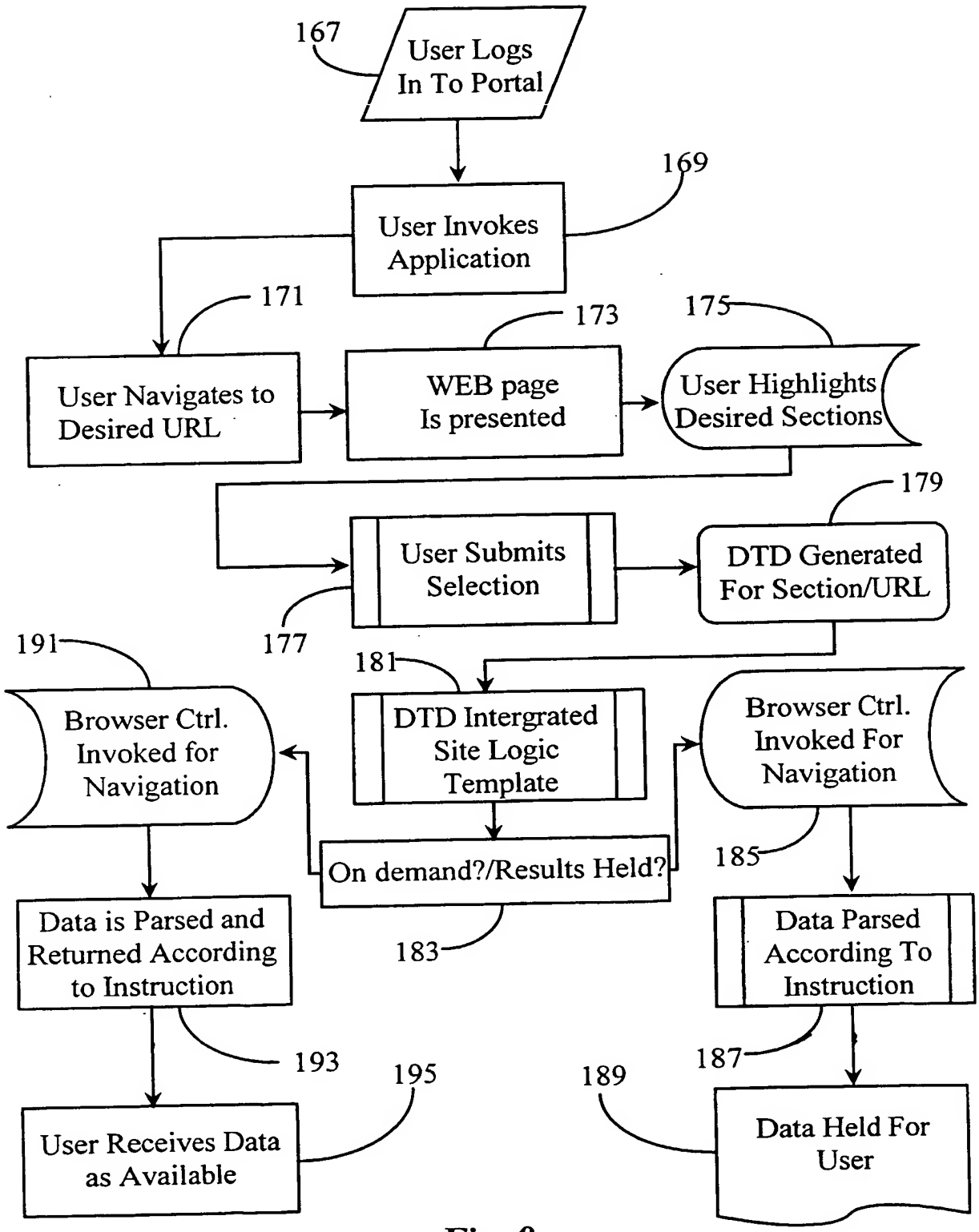

167
User Logs In To Portal
169
User Invokes Application
171
User Navigates to Desired URL
173
WEB page Is presented
175
User Highlights Desired Sections
179
User Submits Selection
DTD Generated For Section/URL
177
191
181
Browser Ctrl. Invoked for Navigation
DTD Intergrated Site Logic Template
Browser Ctrl. Invoked For Navigation
On demand?/Results Held?
185
183
Data is Parsed and Returned According to Instruction
Data Parsed According To Instruction
193
187
195
189
User Receives Data as Available
Data Held For User

Fig. 9

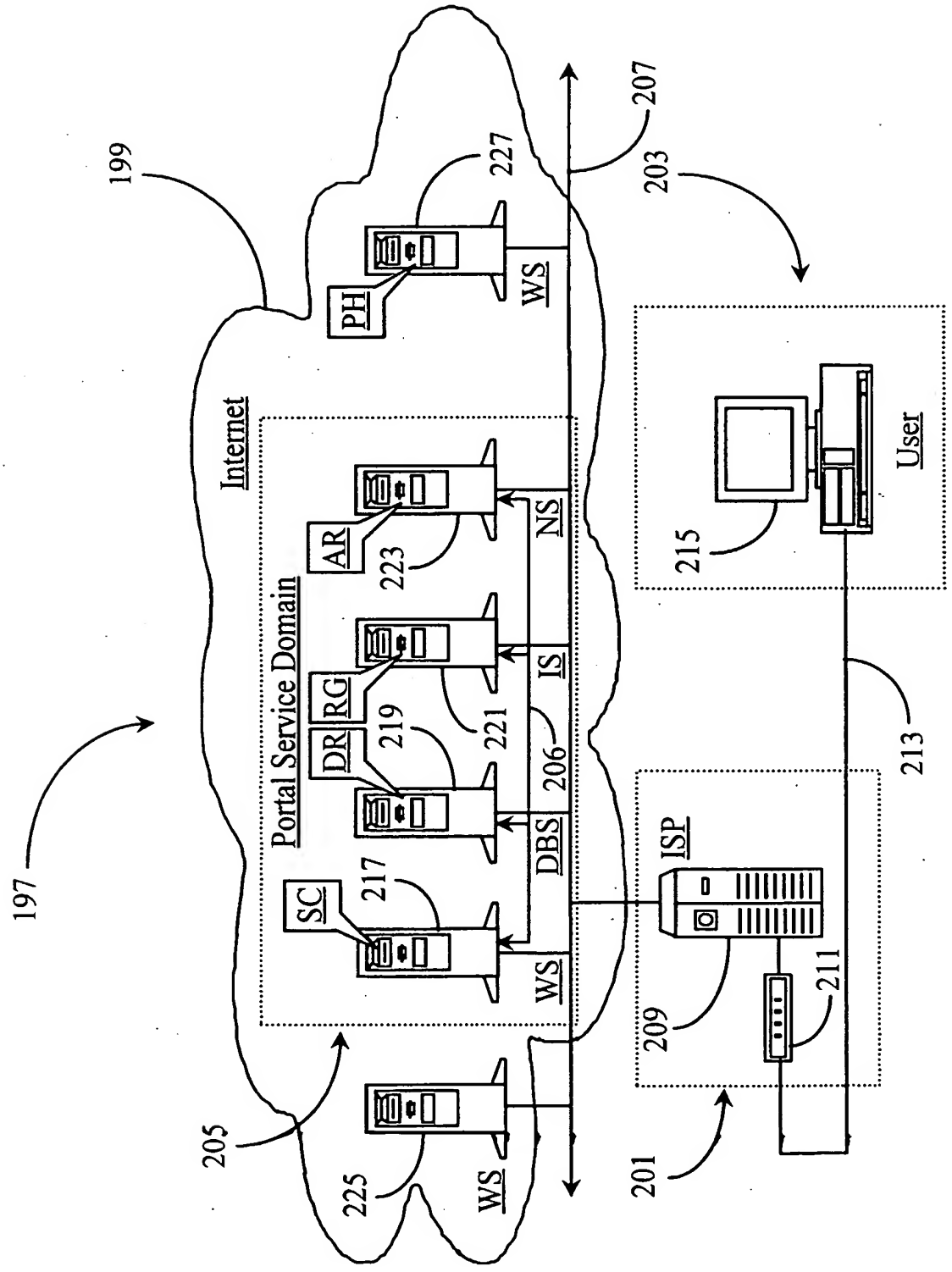


Fig. 10 (NetworkOverview)

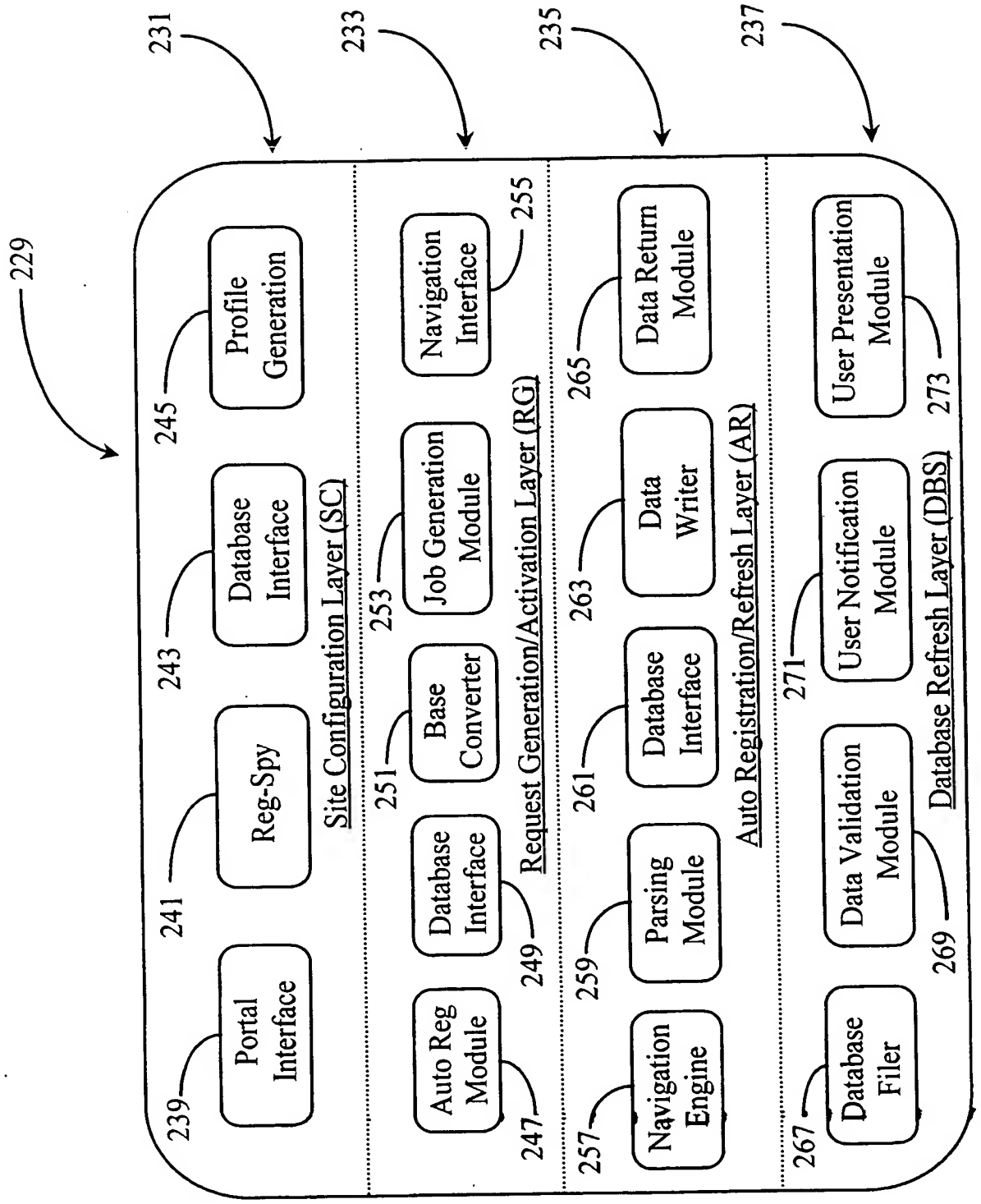


Fig. 11 (SW Architecture)

BNSDOCID <WO. 0180067A1 I_>

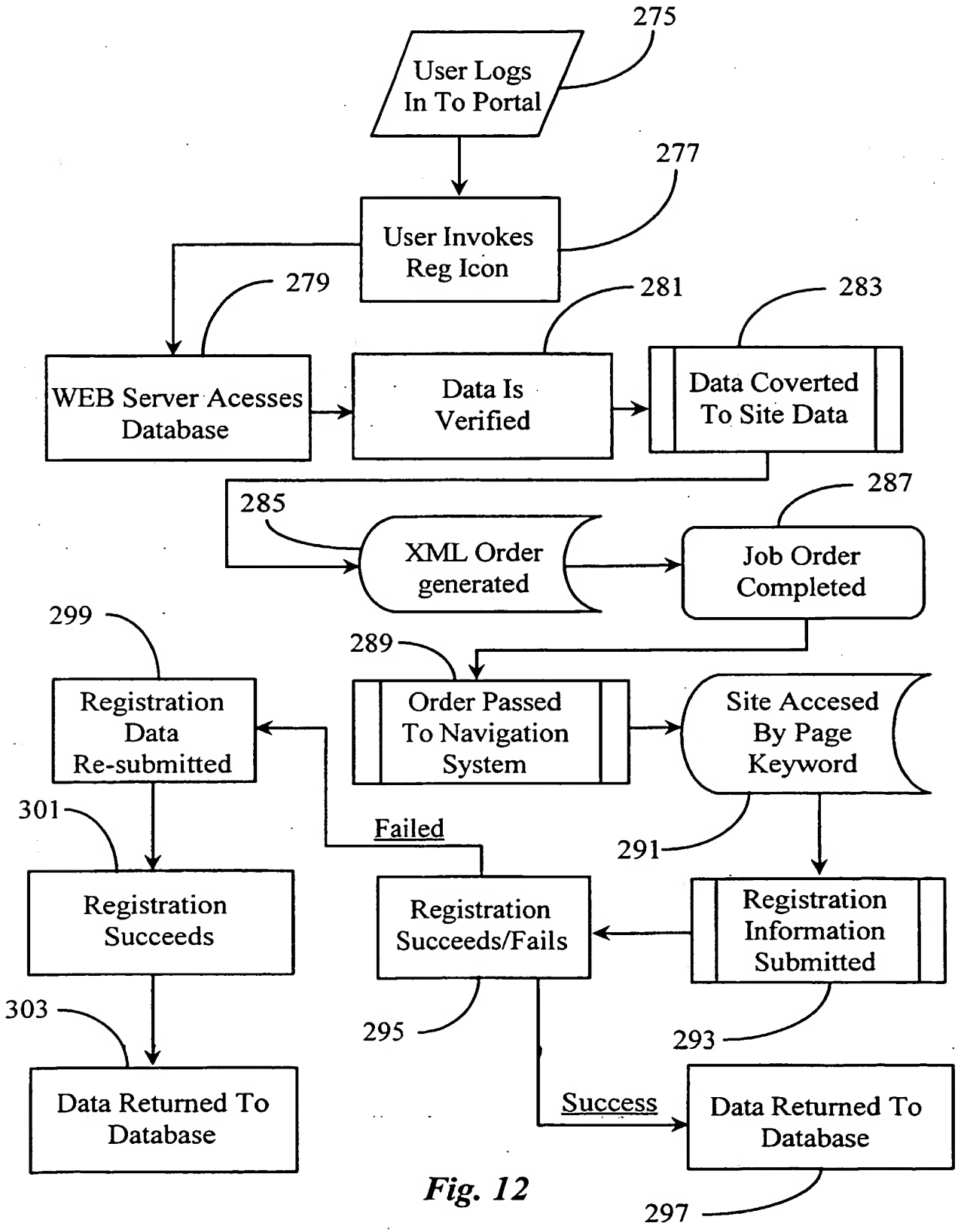

275
User Logs In To Portal
277
User Invokes Reg Icon
279
281
283
WEB Server Acesses Database
Data Is Verified
Data Coverted To Site Data
285
287
XML Order generated
Job Order Completed
299
289
Registration Data Re-submitted
Order Passed To Navigation System
Site Accesed By Page Keyword
301
Failed
291
Registration Succeeds
Registration Succeeds/Fails
Registration Information Submitted
303
295
293
Data Returned To Database
Success
Data Returned To Database
297

Fig. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/08265

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) :G06F 17/00; 15/16
US CL :700/232; 709/218
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 700/232; 709/218

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

ACM
search terms: proxy, electronic forms, data collection

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,884,035 A (BUTMAN et al) 16 March 1999, whole document.	1-18
Y		1-18
Y	US 6,000,832 A (FRANKLIN et al) 14 December 1999, whole document.	1-18
A	CHAUM. D. Security without identification: transaction systems to make big brother obsolete Communications of the ACM. October 1985. Vol 28. Issue 10. pages 1030-1044.	1-18

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
25 APRIL 2001	16 MAY 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer HEATHER HERNDON Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet) (July 1998) ★

BNSDOCID: <WO _____01B0067A1 I >